

香港中文大學研究院教育學部

THE CHINESE UNIVERSITY OF HONG KONG

GRADUATE SCHOOL - DIVISION OF EDUCATION

哲學博士論文

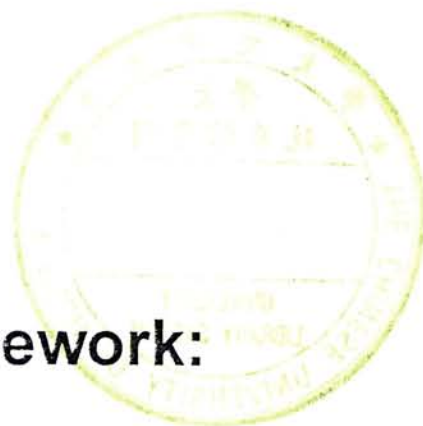
Doctor of Philosophy Thesis

論文題目	Electronic Homework: An Intelligent Tutoring System
Thesis Title	in Mathematics
撰作語言	英文
Language Used	English
研究生姓名	李芳樂
Name of Student	Lee Fong Lok
專修範圍	教育心理學
Specialization	Educational Psychology

論文考試委員會

Thesis Examination Committee

論文導師	Prof. Rex HEYWORTH	夏偉富 教授
Thesis Supervisor		
校內委員	Prof. CHEUNG Ping Chung	張炳松 教授
Internal Examiner		
校內委員	Prof. LO Lam Fat	盧林發 教授
Internal Examiner		
校內委員	Prof. SIU Ping Kee	蕭炳基 教授
Internal Examiner		
校內委員	Prof. CHUNG Yue Ping	鍾宇平 教授
Internal Examiner		
校外委員	Prof. L. GOLDSTEIN	
External Examiner		
校外委員	Prof. CHAN Tak Wai	陳德懷 教授
External Examiner		
學部主任	Prof. CHUNG Yue Ping	鍾宇平 教授
Division Head		



# **Electronic Homework:**

## **An Intelligent Tutoring System in Mathematics**

A THESIS

SUBMITTED TO THE GRADUATE SCHOOL - GRADUATE DIVISION  
OF EDUCATION OF THE CHINESE UNIVERSITY OF HONG KONG

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

By

LEE Fong-lok

31<sup>st</sup> May 1996





# Table of Contents

TABLE OF CONTENTS.....	ii
TABLES .....	vii
FIGURES .....	viii
ACKNOWLEDGMENTS .....	ix
ABSTRACT.....	xi
<b>1 INTRODUCTION.....</b>	<b>1</b>
HOW COMPUTERS CAN HELP OUR CHILDREN .....	2
<i>How Human Tutors Tutor</i> .....	7
<i>Can Computers “Think”?</i> .....	11
<i>Intelligent Tutoring Systems</i> .....	17
ELECTRONIC HOMEWORK.....	18
<i>A Personal Tutor to Students</i> .....	18
<i>The Present Study -- An Investigation into Electronic Homework</i> .....	23
<i>How to Build up Electronic Homework</i> .....	25
<i>Effect of using Electronic Homework</i> .....	29
<i>The Future of Electronic Homework</i> .....	29
CHAPTER SUMMARY .....	30
<b>2 REPRESENTATION OF KNOWLEDGE .....</b>	<b>32</b>
OVERVIEW .....	32
HOW KNOWLEDGE IS REPRESENTED .....	33
SYMBOLIC EXPRESSIONS OR NEURAL NETWORKS .....	36
PROCEDURAL AND DECLARATIVE KNOWLEDGE.....	37
<i>On Evidence Supporting the Procedural- Declarative Knowledge Distinction</i> .....	39
<i>Distinction of Knowledge</i> .....	49
EXPLICIT VERSUS IMPLICIT KNOWLEDGE.....	52
DEGREE OF SOPHISTICATION VERSUS PROCEDURALIZATION.....	53
NOTATION OF KNOWLEDGE .....	59
<i>What Should Be Done But Not What Is Actually Done</i> .....	62
CHAPTER SUMMARY .....	63

<b>3 WHAT KNOWLEDGE TO INCORPORATE AND HOW .....</b>	<b>67</b>
OVERVIEW .....	67
SEPARATE STORAGE FOR DIFFERENT TYPES OF KNOWLEDGE.....	69
DIFFERENT TYPES OF KNOWLEDGE .....	70
<i>The Expert module.....</i>	<i>71</i>
<i>The Student Module.....</i>	<i>78</i>
<i>The Tutoring Module.....</i>	<i>85</i>
<i>The Communication Module .....</i>	<i>121</i>
CHAPTER SUMMARY.....	124
<b>4 PROBLEM COMPLEXITY AND INDIVIDUAL DIFFERENCES</b>	<b>127</b>
OVERVIEW .....	127
COGNITIVE DIFFICULTY OR SIMPLE ITEM DIFFICULTY RATIO .....	129
DIFFICULTY LEVEL OBTAINED BEFORE TEST ADMINISTRATION.....	130
OTHER MEASURES OF PROBLEM DIFFICULTY .....	131
<i>Complexity of Problems .....</i>	<i>132</i>
<i>Problem Complexity Level .....</i>	<i>133</i>
INDIVIDUAL DIFFERENCES .....	133
<b>5 HOW TO IMPLEMENT AND EVALUATE THE SYSTEM.....</b>	<b>136</b>
OVERVIEW .....	136
KNOWLEDGE ACQUISITION .....	140
<i>Expert Module .....</i>	<i>141</i>
<i>Student Module.....</i>	<i>142</i>
<i>Tutoring Module.....</i>	<i>149</i>
<i>Problem Difficulty.....</i>	<i>155</i>
IMPLEMENTATION.....	161
<i>Implementation of Knowledge into Computer Tutor.....</i>	<i>161</i>
EVALUATION.....	162
<i>Formative Evaluation.....</i>	<i>162</i>
<i>Summative Evaluation.....</i>	<i>163</i>
CHAPTER SUMMARY.....	167
<b>6 KNOWLEDGE ACQUIRED .....</b>	<b>169</b>
OVERVIEW .....	169
EXPERT MODULE .....	170



STUDENT MODULE .....	172
<i>Mal-rules</i> .....	172
<i>Strategies for Handling Mal-rules</i> .....	176
<i>Understanding the Errors</i> .....	177
<i>Section Summary</i> .....	209
TUTORING MODULE .....	210
<i>Effects of tutoring</i> .....	210
<i>Scores in Posttest and Ceiling Effect</i> .....	214
<i>Effects of Practice and Tutoring Methods on Retention test</i> .....	214
<i>How Experienced Teachers Perceive</i> .....	221
CHAPTER SUMMARY .....	228
<b>7 PROBLEM DIFFICULTY .....</b>	<b>230</b>
OVERVIEW .....	230
RESULTS OF DIFFERENT MEASURES OF PROBLEM DIFFICULTY .....	231
<i>Students' estimation of Item Difficulty</i> .....	232
<i>Item Difficulty Ratio</i> .....	234
<i>Teachers' Estimation of Problem Difficulty</i> .....	234
<i>Predicted Complexity</i> .....	237
CORRELATION AMONG THE VARIOUS MEASURES OF PROBLEM DIFFICULTY .....	243
<i>How students rate the problems</i> .....	245
PREDICTING THE PROBLEM DIFFICULTY MEASURES .....	246
<i>About the Three Measures</i> .....	249
<i>Practical Considerations</i> .....	252
PROBLEM COMPLEXITY .....	254
USING PROBLEM COMPLEXITY IN ELECTRONIC HOMEWORK .....	258
CHAPTER SUMMARY .....	258
<b>8 SYSTEM EVALUATION .....</b>	<b>259</b>
OVERVIEW .....	259
THE EVALUATION .....	260
<i>Formative Evaluation</i> .....	260
<i>Summative Evaluation</i> .....	270
DISCUSSION .....	288
<i>Who Benefit From Using The System</i> .....	288
<i>Hardware Constraints</i> .....	289

<i>Human-computer interface</i> .....	289
<i>Effect on the use of Electronic Homework</i> .....	290
<i>Expert-Novice Differences</i> .....	292
CHAPTER SUMMARY .....	293
<b>9 CONCLUSIONS AND DISCUSSION .....</b>	<b>294</b>
OVERVIEW .....	294
THEORETICAL ASPECTS .....	295
<i>Why and how do students make errors?</i> .....	296
<i>What makes an expert tutor?</i> .....	302
KNOWLEDGE OBTAINED.....	304
CAN ELECTRONIC HOMEWORK HELP STUDENTS AND TEACHERS?.....	305
<i>Purposes of the Evaluation</i> .....	305
<i>Results of The Evaluation</i> .....	306
SUGGESTIONS .....	306
<i>Machine Learning</i> .....	307
<i>Input Systems</i> .....	307
<i>Better understanding of Human Problem Solving Process</i> .....	307
REFERENCES .....	309
Appendix A: Mal-rule Collecting Tests .....	324
Appendix B: Test on Solving Algebraic Equations .....	334
Appendix C: Tutoring Scripts .....	336
Appendix D: Manipulative Rules Used In Solving Algebraic Equations.....	338
Appendix E: Remediation Rules Used In Solving Algebraic Equations	339
Appendix F: List of Mal-rules.....	341
Appendix G: Teachers' Estimation of Problem Difficulty .....	344
Appendix H: Learning Process Questionnaire .....	349
Appendix I: Questionnaire on the Use of Electronic Homework.....	344
Appendix J: Teachers' Perception on Electronic Homework .....	347
Appendix K: Students' Perception on the Use of Electronic Homework	



in Formative Evaluation.....	346
Appendix L: Results of Students' Perception on Electronic Homework.....	347
Appendix M: Students' Scores in Learning Process Questionnaire	349
Appendix N: Homework 1.....	355
Appendix O: Homework 2.....	358
Appendix P: Students' Retention Test Scores.....	361
Appendix Q: Results of Teachers' Perception on Electronic Homework.....	366
Appendix R: Transcript of Students' Interview.....	368
Appendix S: Installation and Source Code .....	404

# Tables

TABLE 2 .1 THE STEPS INVOLVED IN ADDING FRACTIONS (TAKEN FROM GAGNÉ, 1985). -----	55
Table 2 .2 Production Representations For the First Three Steps in Adding Fractions-----	56
Table 2 .3 A Production for Computing the Least Common Denominator -----	57
Table 3 .1 Production Rules Used in Young & O'Shea's Production System for Subtraction	91
Table 3 .2 Trace of Production-----	92
Table 5 .1 Time Frame for the Implementation and Evaluation Process -----	137
Table 6 .1 Table showing groups of mal-rules obtained-----	174
Table 6 .2 Dialogue showing that a student treated "log A" as "log times A"-----	181
Table 6 .3 Dialogue showing the reason why a student treated "log AB" as "log A times log B" -----	184
Table 6 .4 An Extract of the protocol of student S2 on log A as log times A. -----	185
Table 6 .5 Extraction of Protocol of student S3 to illustrate the confusion caused by properties of logarithms. -----	190
Table 6 .6 Protocol of Student S3 that attempted error [E2] -----	202
Table 6 .7 List of mal-rules as explained by meta-rules -----	206
Table 6 .8 Mean Duration (in minutes) by Tutoring Conditions and Practice Conditions	210
Table 6 .9 Test Results by Tutoring Conditions (N=50)-----	211
Table 6 .10 Mean Scores by different practice and tutoring conditions in retention test--	214
Table 6 .11 Analysis of scores in retention test by practice and tutoring methods with -pretest as covariate -----	215
Table 6 .12 Analysis of Scores of Subjects without Practice in Retention test by Tutoring Conditions with Pretest as Covariate -----	217
Table 6 .13 Causes of errors and ways to remedy them as perceived by experienced teachers-----	222
Table 6 .14 Table showing Relation Between Meta-rules and Reasons Given by Teachers	227
Table 7 .1 Table showing various Mean Measures of Item Difficulty-----	233
Table 7 .2 Statistics of Teachers Contributed to the Estimation of Problem Difficulty ---	236
Table 7 .3 Teachers' rating on importance of factors affecting problem difficulty-----	237
Table 7 .4 Problem Difficulty as Predicted by the Difficulty Factors -----	240
Table 7 .5 Correlation Coefficients Among the Measures of Problem Difficulty-----	244
Table 7 .6 Summary of multiple regression coefficients found-----	248
Table 7 .7 Variables in the Equations to Predict the Problem Difficulty Measures-----	251
Table 7 .8 Summary of Multiple Regression Analysis for Variables Predicting Students' Prediction of Problem Difficulty-----	255
Table 7 .9 Correlation Among the Problem Difficulty Measures -----	256
Table 8 .1 Summaries of Homework Scores-----	272
Table 8 .2 Table showing the Mean Retention Test Scores of the Participated Schools --	278
Table 8 .3 Analysis of Variance for the Retention Test Scores (Rttot) -----	280
Table 8 .4 Analysis of Variance for the retention test score with Sch=2 -----	282
Table 8 .5 Analysis of Students' Perception of Electronic Homework -----	285
Table 8 .6 Analysis of Teachers' Perception on using Electronic Homework -----	287

## Figures

Figure 1 .1 An Example of Question used in Physics Topics -----	4
Figure 3 .1 An Overview of the structure of Topics in Chapter 3.-----	68
Figure 3 .2 An Example of Student Interaction in using PIXIE -----	75
Figure 3 .3 An Example of Student-Computer Interaction in LISPITS-----	76
Figure 3 .4 A Procedural Network for Subtraction -----	87
Figure 5 .1 Structure of Electronic Homework-----	138
Figure 8 .1 Student-computer Interface in Electronic Homework-----	267



## ACKNOWLEDGMENTS

This dissertation would not have been possible without the support of the principals, teachers and students of the schools participating in the evaluation of the computer system. I wish to extend special thanks to the following teachers, some of whom gave up their valuable time during normal lessons to allow the evaluation to take place, some helped with the installation of the computer system and others coordinated the whole evaluation exercise within their schools:

Mr. CHAN Wai-man, Raymond	Henrietta College
Mr. CHAN Kam Ming	Nam Wah Catholic Secondary School,
Mr. CHU Chung-pok	Po Leung Kuk Lee Shing Pik College
Miss HO Mei-fan	St. Paul Conventional Secondary School
Mrs. LAM LO, Fung, Lydia	St. Paul Conventional Secondary School
Mr. LAW Huk-yuen	Henrietta College
Mr. LEUNG Ka-wing	Pui Ching Middle School
Mr. LI Yun-key, Roy	St. Paul Conventional Secondary School
Mr. LI Kwok-ning	T. W. G. Hs. Chang Ming Thien College
Mr. LUNG Tak-yee	Po Leung Kuk Lee Shing Pik College
Mr. NG Kwok-on	Nam Wah Catholic Secondary School,

## Acknowledgments

Mr. YIP Chi-tim

Pui Ching Middle School

To my colleagues in the Faculty of Education of The Chinese University of Hong Kong who contributed valuable ideas and even taking over some of my duties to enable me to complete this project on time, my heartfelt gratitude.

My thanks also go to my five long-suffering students, Miss TONG Choi-ieng, Miss CHENG Chui-king, Mr. HUNG Hing-yuen, Mr. SAU Hang-chor and Mr. MAN Ho Kei. They tirelessly and cheerfully transported piles of questionnaires and boxes of equipment between the Faculty and the schools, often going without lunch, to carry out the experiment. My thanks are also due to Mrs Grace CHEUNG for going through all the papers and giving valuable suggestions.

Without the effort of the advice and assistance of members of my Supervision Committee, Dr SIU Ping-kee, Dr LO Lam-fat and Dr CHEUNG Ping-chung, this work would not have been completed. My sincere thanks go to them. I would like to pay special tribute to Dr SIU Ping-kee, Supervisor of my Master's thesis and now a member of my doctoral Supervising Committee. It was due to Dr SIU's encouragement and unflagging enthusiasm on my behalf that I had the courage to embark on this venture.

Most of all, I would like to thank my supervisor, Dr Rex HEYWORTH for being so patient with my misconceptions and for directing me unerringly through all the difficulties along the way.



## Abstract

Homework is an important component of the learning process for it helps students to consolidate what they have learned in the non-regimented atmosphere of their own homes. However, if students were to work alone on their homework assignments, they would be liable to make mistakes which, if not noticed, would be repeated several times throughout the assignment. In order to minimize the chance that repeated errors would become systematic errors, Electronic Homework was designed as a computer system with the ability to diagnose errors and remedy them as soon as they occur. Besides being a computer tutor, Electronic Homework was also designed to function as a teacher's assistant to provide not only information on students' errors but also to help teachers to mark assignments.

The present study focused on a series of investigations into why and how students make errors, how teachers help their students to correct the errors and how problems should be ordered so that students can learn more effectively. The ultimate aim was to discover how the computer system could be of benefit to students.

Artificial intelligence techniques were employed to store the knowledge required by this computer system. Errors made by students when solving logarithm problems were collected and six students were interviewed. From the data collected, a new type of error, referred to as the misperceived error, was suggested. Also, a model as a simple explanation for this type of error was offered. It was found that, in terms of representing errors in computer systems, errors can be categorized and that each

category can be described in terms of one meta-rule. The use of meta-rules helps to simplify the structure of the computer system.

The study looked into teachers' diagnoses of student errors and their strategies to help their students remedy these errors. Analysis of the data revealed that teachers helped their students with strategies based on their diagnoses of the errors and the diagnoses generally agreed with those described by the meta-rules.

The problem of how to order the exercises was studied by asking teachers and students to estimate the difficulty levels of a series of problems. These estimated difficulty levels were found to be predictable by several factors, such as number of steps required to complete the problem, number of operations in the problem expression and the degree of familiarity with the problems. From these findings, a measure of the problem difficulty was developed.

Finally, having incorporating the knowledge obtained, the computer system was tested with students and teachers selected from six secondary schools in Hong Kong. Results of the evaluation showed that the computer system was able to give assistance to students in one of the participating schools in solving less abstract problems. The reason why this occurred in only one school is not known but one possible reason is that the students are more motivated towards learning than the other students in the test. Areas for further study derived from the research findings are then suggested.

# ***CHAPTER 1***

## **INTRODUCTION**

Increasing social prosperity and the lowering prices of computer hardware mean that micro-computers are now becoming more and more popular not only in Hong Kong but the world over. Although there are no statistics to show how many families in Hong Kong own micro-computers, a small scale survey done by Lai (1995) on micro-computer usage among secondary two (grade 8) students in a government school<sup>1</sup> in Hong Kong reveals part of the picture. According to Lai, 103 (84%) of a total of 122 students reported that they know how to operate computers. It is true that some of the students viewed computers as sophisticated toys but a small proportion of them (14 out of 122) indicated that they used commercially available software<sup>2</sup> to assist them in their studies. The sample school is located in a public housing estate and its roll is composed mainly of students of the lower socioeconomic class. This survey is a reflection of the popularity of computers among children in Hong Kong, and also reveals that some students are now beginning to use computers as learning tools. Although some of them may not yet be in a position to own computers, they may

---

<sup>1</sup> Government schools in Hong Kong are schools wholly funded by the government. Although nearly all schools in Hong Kong are subsidized by the government, most of the schools are called subsidized schools with a small proportion of their expenditure to be responsible by the organizations who run the schools. However the difference in financial situations does not mean any difference in academic standards.

<sup>2</sup> The software used for learning were not reported. However there are quite a number of educational systems available in Hong Kong although contents of these systems are not always related to the school curricula.



consider purchasing one in the near future. Students will then benefit from using computer tools which can help them to learn.

# How Computers Can Help Our Children

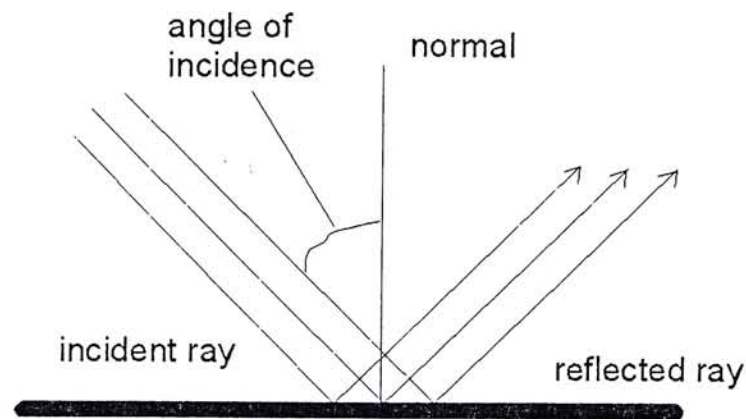
Computers have been used in education for more than thirty years (Mandl & Lesgold, 1988). Earlier systems using computer technologies in instruction were called computer-assisted instructional systems. In such systems, students are normally given a short piece of didactic material to study and this was followed by a test. Students' responses determined further action to be taken -- whether they were allowed to go on to study other pieces of materials or whether they were required to study further materials to correct misconception. All instructions to students, decision points on which subsequent operations were to be taken, questions and their possible responses had to be prespecified (Goodyear, 1991). This way of incorporating knowledge into a computer system was described as inflexible (Mandl & Lesgold, 1988) in the sense that any decision points in the system were for specific lessons and any possible responses were for specific questions.

There was no knowledge item that could be shared among the different parts of this kind of systems. An example is "Computer works" (MYTHOS software Inc., 1993) designed to teach the different parts of a computer. There are eleven lessons in all including those on computer systems, disks and storage, etc.. Each lesson sequentially presents information on different parts of a computer by using graphics and texts. After each lesson, a quiz of ten questions tests whether the students have mastered the knowledge. Students who answer each question correctly are allowed to

proceed to the next question while each incorrect answer is followed by further instruction. Thus in this system, questions and instructions are closely linked to the lessons, and cannot be applied to any other. Even if there are knowledge pieces that are common to two lessons, the technology of the system requires that they be incorporated separately. This kind of knowledge incorporation would require a large amount of human labor as well as a large computer memory space. Detailed discussions on this will be given in later sections.

The major shortcoming of this kind of traditional computer-assisted instructional system is that they cannot understand students' responses. In such systems, students' performances can only be reflected through figures such as the number of items correctly answered and it is impossible to detect why and how the students made the errors. "Computer Works" is one example of systems in which students' performances are represented by the number of questions they answer correctly in the quizzes. A similar example is a system called "Physics Topics" (1995) which is intended to be a tutor for various topics in secondary school physics. For each topic, texts and pictures illustrate the content to be taught followed by a multiple choice test for evaluating performance. An example of such test items can be found in Figure 1.1 below:





If a ray of light strikes the surface of a mirror at a right angle to the mirror, what is its angle of incidence?

1. 0 degree
2. 45 degree
3. 90 degree

Figure 1.1 An Example of a Question used in Physics Topics

For every such test item, students are required to pick the correct answer from a number of choices. Only statistics on the numbers of items correctly answered are provided in the system, and there is no further analysis of the students' responses. What the system can do is to give a report on a student's performance, or in the words of Sleeman and Brown (1982), a student's performance is evaluated based more on parametric summaries of behavior<sup>3</sup> than explicit representations of his knowledge.

---

<sup>3</sup> Parametric summaries of behavior are ways of representing behavior by using numerical values. Describing students' performances by using number of correct responses is one example. Other examples include describing students' performances in terms of the length of time used, the number of errors made, etc..

Parametric summaries alone may give a rough picture of the students' performances but cannot provide other information such as why and how students made the errors. Although there may be feedback given for incorrect responses, these responses are prespecified in a way that the designer of the system predicts that an error would occur at a particular point of the tutoring or testing process. He or she therefore puts in a piece of knowledge specifying what response (or feedback) should be given when this error occurs. The following example might make this clearer:

When solving the algebraic equation:

$$[E11]^4 \quad x+2=5,$$

a system designer predicts that some students might give the answer:

$$[E12] \quad x=5+2$$

He or she therefore instructs the system to give a certain response whenever the equation [E12] is detected. Note that as in traditional computer-assisted instructional systems, instructions have to be incorporated for different questions. Thus even for a similar equation:

$$[E13] \quad x+3=7$$

and a similar error:

$$[E14] \quad x=7+3$$

---

<sup>4</sup> This convention will be used throughout the whole manuscript to represent different expressions. The first letter inside the bracket represents whether this is an equation(E), a rule(R) or a formula(F), the number following this letter represents what chapter this expression is in and the final number represents the order of this expression in the chapter. Thus [E11] means this is the first equation in chapter 1.

another instruction still has to be incorporated even though the content of the instructions given to [E12] and [E14] may be identical.<sup>5</sup>

It cannot be said that systems designed in this way understand students' errors since all they can do is follow the designer's instructions. Whether this system is a good tutor or not will very much depend on the designer's ability both in diagnosing and remediation of errors. The system is no more than a collection of actions that have to be taken.

Even if one such system is well designed, it is still restricted by many practical considerations. To provide diagnoses and remediation to students, the system designer has to think of all the possible intermediate steps and answers that may be given to students and then devise plans to help to correct those that are wrong. All these have to be stored into a computer system. The more such possible routes and remediation steps are incorporated, the higher will be the ability of the system in diagnosing and remediation of students' errors. In this case, it will require a large storage space in the computer. If, in addition, these systems are to be used in the microcomputers readily available to our students which are less expensive and therefore less sophisticated, the consequence will be: either we have to accept systems with fewer pre-specified responses and instructions and lower diagnosing and remediation abilities, or we have to confine the system to narrow subject areas so that fewer responses and instructions are required. Neither these options is desirable if computer assisted instruction is to play an important role in teaching.

---

<sup>5</sup> In most cases, the instructions are always done on final answers given the students, and the feedback given is just telling the students that they are wrong without any diagnoses on the causes of the errors.



## How Human Tutors Tutor

If we compare the above computer instructional process with the way a human tutor operates during the same process, we may gain valuable insight. Human tutors do not always use specified patterns of reactions and instructions. For typical errors, an experienced tutor may react quickly and accurately. An example is when a sixteen year old student gives 6 for the simple arithmetic problem  $2+3=?$ , the experienced tutor might immediately point out that it is a careless mistake since there is no reason for a student of his or her age to make such an error. However, for most other cases, the tutor may need to deliberate more carefully. He or she might have to analyze the student's work before deciding on the action to be taken. Basis for the analysis would be the tutor's own domain-specific knowledge<sup>6</sup> and pedagogic knowledge.<sup>7</sup> The following example serves to illustrate this process:

When a student is asked to simplify a logarithmic expression:

$$\log 5 =$$

and he is told that  $\log 2 = 0.3010$  and  $\log 3 = 0.4771$ . This student responds by giving:

$$\log(2 + 3)$$

The tutor responds by saying "correct". The student then writes:

$$\log 2 + \log 3$$

---

<sup>6</sup> Domain-specific knowledge is the knowledge on the subject concerned. For example, if a student is solving mathematics problems, then the domain-specific knowledge is mathematics knowledge.

<sup>7</sup> Knowledge on how to teach.

The tutor might notice that the student is using a rule<sup>8</sup> in the form of:

[R11]<sup>9</sup> IF an expression in the form of  $\log(\text{Number1} + \text{Number2})$  is to be simplified,

THEN express it as  $\log(\text{Number1}) + \log(\text{Number2})$

where *Number1* and *Number2* represent two numbers. The reason for this error is the misconception that logarithm of a number is identical to the product of a quantity called "log" and that number. In terms of a rule, this can be written as:

[R12] IF an expression  $\log \text{Number}$  is to be simplified,

THEN treat it as the product of a variable log times *Number*

Note this is a reasonable assumption on the part of the student since he or she is used to expressing the product of two variables *a* and *b* as *ab*, and the product of a number 2 and a variable *x* as *2x*. To some students, expressions involving the same two quantities but which differ only by the presence or absence of multiplication sign between them, are actually the same expressions.

By using rule [R12] and the distribution law which states that:

[R13] IF an expression in the form of  $a \times (b + c)$  is to be simplified,

THEN express it in the form of  $ab+ac$

where *a*, *b*, *c* are any quantities. The result then follows.

---

<sup>8</sup> A rule is the description of a knowledge piece on what should be done under certain circumstances. For example, the rule [R11] describes when an expression in the form of  $\log(\text{Number1} + \text{Number2})$  is to be simplified, it should be expressed in the form  $\log(\text{Number1}) + \log(\text{Number2})$ .

<sup>9</sup> By using the same convention as in note 4, this means the first rule in chapter 1.



Having identified the cause of the error, the tutor then instructs the student that the logarithm of a number is not identical to a quantity called "log" times the number and that rule [R11] is wrong. The student may then be shown the correct way of doing this problem.

Notice that the tutor decides the instructional action based on the knowledge of mathematics ([R11] and [R13]) and pedagogic knowledge viz. the misconception should be clarified. The tutor does not have to go through the above process every time an identical error is detected. Having acquired enough experiences, the tutor might store the knowledge as a formal rule:

[R14] IF student expresses  $\log(\text{Number1} + \text{Number2})$  as  
 $\log \text{Number1} + \log \text{Number2}$   
  
THEN instruct the student that  $\log(\text{Number1} + \text{Number2})$  is not  
 $\log \text{Number1} + \log \text{Number2}$   
  
since  $\log(\text{Number1} + \text{Number2})$  is not  $\log \times (\text{Number1} + \text{Number2})$

where *Number1* and *Number2* are any two integers. By using this knowledge, the experienced tutor can now diagnose faster.

On the other hand, if the above process is to be done by traditional computer-assisted instructional programs, it may have to store the following rule:

[R15] IF student expresses  $\log(2+3)$  as  $\log 2 + \log 3$   
  
THEN instruct the student that  $\log(2+3)$  is not  $\log 2 + \log 3$

since  $\log(2+3)$  is not  $\log \times (2+3)$

Notice that the variables *Number1* and *Number2* in [R14] are now replaced by exact numbers. If there is only one problem to be solved, there should be no difference when either one of the rules is applied since both would yield the same results. However, if there is one more problem such as  $\log(4+5)$  to be diagnosed, one more rule [R16] has to be added:

[R16]    IF            student expresses  $\log(4+5)$  as  $\log 4 + \log 5$   
  
          THEN    inform the student that  $\log(4+5)$  is not  $\log 4 + \log 5$   
  
                     since  $\log(4+5)$  is not  $\log \times (4+5)$

Even if one can replace the action parts (the part of the rule behind the word THEN) of [R15] and [R16] with that in [R14], the three rules [R14], [R15] and [R16] are still different in their condition parts (the parts between the words IF and THEN).

In normal learning processes, students are usually required to practice solving many problems for which many knowledge pieces are either identical or similar. Traditional computer-assisted instructional systems do not try to identify these similarities but just store a separate set of rules for each problem even if two or more problems are identical. The number of rules required would thus explode very quickly when the number of problems increases. However, a human tutor would use just one rule to include all similar ones (for example, the rule [R14] includes the rules [R15] and [R16]) and is thus more efficient and economical in handling the knowledge required.

Human tutors are thus characterized by generating the actions in real time or by using more efficient rules. They can use much less memory than traditional computer systems to store the knowledge. If the knowledge of human tutors can be extracted and fed into a computer system, this new computer system would not be so limited by the constraints imposed by the hardware and can then handle more complex problem-solving processes. It is possible that such a computer system can replace a large part of the human tutor's work.

### **Can Computers "Think"?**

Traditionally, computer systems are algorithm<sup>10</sup> oriented. In programs written with conventional computer languages such as FORTRAN, Pascal, Basic or C, often the first thing you will need is an algorithm (a way of doing it) to achieve the task. Then you can design a sequence of steps based on this algorithm. The task is then achieved by following the specified steps. We cannot find any sign of decision-making in this kind of computer program since all it can do is to follow human instructions.

On the other hand, the invention of computer languages called Prolog and Lisp provides a completely different programming environment. In writing a program in Prolog, for example, there is no need to specify an algorithm. Instead, the Prolog programmer asks what formal relationships and objects occur in his problem, and what relationships are "true" about the desired solution (Clocksin & Mellish, 1987). A

---

<sup>10</sup> An algorithm is a way of doing things which will ensure the correct answer. A simple example is to sort (arrange) two names in alphabetical order. The algorithm would be first comparing the two names, if the first name is of lower or the same order of the second, then stop; if the first one is of higher order then interchange the two.



Suppose it is further given that a person A likes person B if A likes a third person C and this C likes B. This knowledge will then be added to the system in the form of a rule:

[R17] Person A likes person B if A likes person C and C likes B.

If the same goal is to be achieved after adding this new knowledge piece, the answers would then be John, Joan and Arthur. While the first two names are directly obtained from the facts, the last name is deduced from the fact that Mary likes John and John likes Arthur. The fact that Mary likes Arthur is not stored in the program but is inferred when the program is being executed. A Prolog program is thus characterized by this type of inferring abilities. Also, there is no specific algorithm to guide the system to find the answers. One can easily change the answer by just changing the knowledge in the system. For example, if the rule [R17] above is changed to [R18]:

[R18] Person A likes person B if A likes person C and C likes B and A does not dislike B.

then the answer becomes John and Joan again; Arthur is not in the list since Mary dislikes him although Mary likes John and John likes Arthur. Thus the answer is rather inferred from the knowledge in the system than determined by the algorithm specified. It is this inferring ability which enables such computer systems to be considered as "can think" since they can now generate their own solutions without strictly following the ways specified by human beings.

## Introduction

Prolog program thus contains known facts and relations between facts. The goal of a program<sup>11</sup> can sometimes be achieved by information supplied by the programmer, but most of the time, it is achieved by inferring from the known information. The following simple example can be used to illustrate how Prolog achieves its goals:

### A Simple Prolog Program

A Prolog program will require facts. Suppose now it is known that:

Mary likes John<sup>12</sup>

John likes Arthur

Ann likes Susan

Arthur likes Ann

Mary likes Joan

Mary dislikes Arthur

Arthur dislikes Mary

Now the goal is to find out whom Mary likes. The system would search among the facts that can match a question in the form of "Mary likes \_\_\_\_" and reports those which do. In this case, the answer would be John and Joan.

---

<sup>11</sup> The goal of a program is what the program is designed for. For example, a program may be designed to find out the tallest boy in the class, then the goal of the program will then be finding the name and the height of the tallest boy among the list of boys entered.

<sup>12</sup> The actual syntax in Prolog is in the form of `like(mary, john)`. For simplicity, it is not used in this chapter but will be used whenever appropriate.

### **Artificial Intelligence**

With the help of languages such as Prolog and Lisp, computer systems can now carry out many intelligent functions that were originally by human functions. This kind of studies of computations which makes it possible for a program to perceive, reason, and act is called Artificial intelligence (Winston, 1992).

If artificial intelligence techniques enable the development of computer systems that can simulate human thinking, it might also be possible to develop a computer system that can teach like a human tutor provided that the essential properties of human tutors can be identified and incorporated. The following sections explore ways of doing this.

### **How Machines “Think”**

Machines, just like human, need two components to “think”: the knowledge and the ways of using the knowledge. For example, a student knows that all numbers divisible by 2 are called even numbers, and because the number 8 is divisible by 2, he deduces that 8 is an even number. In this case, the student has the knowledge represented by the following rules:

[R19]        If        *Number* is divisible by 2  
  
              Then    *Number* is an even number.

[R20]        8 is divisible by 2

The result that 8 is an even number is then inferred. Notice that originally, the result is not a part of the student's knowledge base, it is the inference mechanism



within the student's mind that produces it. Hence for a machine to “think”, it should also be provided with the required knowledge and an inference mechanism.

### **Knowledge Representation**

On knowledge representation, several techniques have been developed in the field of Artificial Intelligence. For example, in building a computer tutor for solving ‘crane boom’ problems, which typically include a beam (the boom) attached to a wall and supported at the other end by a cable attached to the same wall. Concepts in the domain are represented by frames (Woelf, 1987). In this case, a frame is an explicit set of attributes each with a default value. Mao and Lin (1992) used skill graphs<sup>13</sup> to link up the skills involved in symbolic calculation into a hierarchy.

For mathematics knowledge, a common convention is to represent the knowledge pieces in the form of if-then rules such as [R19] shown above. Such rules are called productions and a set of all productions is called a production system.

### **Inference Ability**

To establish the ability to infer from any given knowledge, recently developed computer languages, called artificial intelligence languages, are incorporated with these inference abilities. The most commonly used ones are called Lisp and Prolog. Lisp is the short name for "List Processing" where a list is a sequence of numbers or names. New knowledge can be inferred from knowledge stored in lists. Prolog is the name given to represent "Programming in Logic". As the name implies, programs written in Prolog are similar to logic. Knowledge is stored in the form of If-then rules

---

<sup>13</sup> A skill graph is a graph linking up all the required skills to solve a particular problem.

## Introduction

called predicates and there is an implicit mechanism to do the inference. Given a problem, Prolog solves it by searching through its knowledge base for the appropriate set of rules to achieve the goal. The following simple example serves to illustrate how Prolog solves problems, where the rule [R19] is identical to one shown in above paragraphs.

Suppose the system contains the following two rules:

[R19] IF *Number* is divisible by 2

THEN *Number* is an even number.

[R21] IF the remainder is 0 when *Number* is divided by 2

THEN *Number* is divisible by 2.

where *Number* represents any real number. If the task is to show the number 64 is an even number, Prolog first sets this as the goal then searches for conditions that this goal can be achieved. In this case, rule [R19] is found and the condition will be that 64 must be divisible by 2. Prolog then sets up a goal to find the condition that 64 can be divisible by 2. Rule [R21] is then found. Assuming it can be shown that when 64 is divided by 2, the remainder is 0, and there are ways to find the remainder, then rule [R21] is proved and eventually rule [R10] is also proved. The initial premise that the number 64 is an even number is now proved. It should be noted that there is no such knowledge in the system to show that 64 is divisible by 2. The knowledge is deduced by other knowledge stored and the inference mechanism. This mechanism is able to make some systems act like human beings to generate real time responses under novel

situations. A system written in a language like Prolog would be one such example. It should be noted that this kind of inference does not have to be done by artificial languages like Prolog, other computer languages like C and BASIC can as well do the same job, provided the developer can incorporate the inference mechanism into the system. With Prolog, this is inherited from the language and is thus easier.

## **Intelligent Tutoring Systems**

With the artificial intelligence techniques, we can now try to simulate the human tutoring process. Systems that simulate the way a human tutor helps students learn are called Intelligent Computer Assisted Instructional systems (ICAI), Intelligent Tutoring Systems (ITS), or Intelligent Learning Environments (ILE). The basic difference between these systems is the amount of initiative required from student input. The first two systems are often regarded as similar and were defined as computer programs that use AI techniques for representing knowledge and are capable of carrying on interaction with a student (Sleeman & Brown, 1982). They are instructional systems that emphasize guiding students through the learning process, while Intelligent Learning Systems expect students to take more initiative in choosing the different learning tools provided. An interesting example of Intelligent Learning Environments is the Learning Companion Systems designed by Chan & Baskin (1990). In this learning environment, besides the usual characters of the student and the computer tutor, there is an other actor called the learning companion. The learning companion is a computer simulation of another student and its role is to learn with the student through exchange of ideas and collaboration. Hence learning in this



environment is not confined only to that guided by the tutor but also includes the experience of working with others.

# Electronic Homework

Homework is an effective means of helping students to consolidate what they have learned. Traditionally, this is done at home with pencil and paper. When they have problems, some lucky students may have immediate assistance from their parents, siblings or others who act as human tutors. However, most of them would have to wait until the next day before they can ask for the teachers' help. But at the time they are doing their homework, they would have to resort to other means of overcoming the difficulties which, if unresolved, may become sources of later errors (VanLehn, 1990, 1982b). It would be desirable if a personal tutor could be made available to help each student when he or she encounters homework problems.

## A Personal Tutor to Students

A personal tutor should be able to help students of varied abilities. For better students, immediate feedback should be provided to strengthen the learning effect. For students of lower ability, misconception or errors should be corrected immediately before they become stable errors<sup>14</sup> (VanLehn, 1990). If they do not know how to continue, immediate help should be provided. Hence students' correct behavior will be reinforced and incorrect behavior avoided. Ideally, these should be done by human

---

<sup>14</sup> Stable errors are errors that happen systematically, they are thus also called systematic errors. Cognitive psychologists (for example, VanLehn, 1995; Sleeman, 1985) think that stable errors initially happen occasionally but happen more and more when the corresponding rules are used more.

tutors. However, if human tutors are not available, computer tutors that can act like human tutors may be the next best solution.

To provide a computer tutor to each student is easier said than done. The necessary condition is that each student should have a personal computer at home, a condition which is not fully met for the time being. However, with the increasing prosperity of society and the lowering of the prices of personal computers, it is believed that in the coming few years, this condition will be satisfied.

### **Lack of Suitable Software for Helping Students with Their Homework**

The biggest handicap is the lack of suitable software. Although currently there are quite a number of educational systems available (examples shown in the next sections), they are not suitable for the present purpose. The reasons, according to the order of significance, are their incompatibility with school curricula and the language of instruction they are using. The following sections discuss these in more detail:

#### **Incompatibility with School Curricula**

Among the available educational systems, there are not many which can be directly used in the present school curricula. A lot of them are information providers like dictionaries or encyclopedias. For example, the American Heritage Talking Dictionary by Softkey International Inc. (1994) is a software that can actually pronounce the words in a human voice besides giving the meaning. These are useful learning tools, but surely cannot actively guide the students in doing their homework.

Another type of system that can be used for educational purpose is a popular spreadsheet software called Excel (Microsoft, 1995). It has been reported that Excel can be a useful tool in teaching such concepts as electrical circuits and heat transfer in physics (Silver, 1994; Drago, 1993). A spreadsheet is somewhat like a table of cells. You can assign certain cells to represent independent variables and others as dependent variables. By entering values for the independent variables and relationships among the independent and dependent variables, values for the dependent variables can be calculated automatically. More than that, whenever any values of the independent variables or the relationships among the variables are changed, all other related values will be automatically updated. Hence, spreadsheet is a convenient tool for testing hypotheses or simulating physical situations, which makes it a very useful educational tool in training creative thinking.

Although spreadsheet systems are useful, it is also clear that their use is limited to the testing of hypotheses or simulating physical situations. When it comes to mathematics homework, especially in training skills like solving equations, such systems cannot offer much help.

### **CAI Programs**

Educational systems that are related to school curricula seem to function at two extremes: restricted or loose. Some programs like Advanced Physics (Scientia, 1995) would include illustrative pictures and even animation for the teaching of concepts and ideas. But to promote interaction between students and the computer tutor, there are only short questions or multiple choice items to test whether or not the students have mastered the concepts. If students answer incorrectly, they will be led to



prespecified paths to discover their own misconception and are then expected to correct their own errors. Thus, the help offered to students by this kind of systems is limited.

### **Drill-and-Practice Programs**

The Advanced Physics program mentioned above is an example of a restricted program in the sense that students are expected to follow paths prespecified by the system designer. There are also other systems like Mathematics Review (Education Time Courseware, 1995), most of whose capabilities emphasize on drill-and-practice. Students are given exercises to practise. The practice items are either in the form of multiple choice or short questions for which only short answers are expected. The purpose of this software is to drill the students until they can reach a certain degree of competence which is measured by a test similar to the exercises. Those who pass the test will be allowed to go on to the next part of the system. Students who cannot pass the test will be asked to either revise certain parts of the materials or be given additional materials to read until they master the subject. There is no attempt to understand students' errors and all remedial measures are prespecified.

### **Simulation Programs**

At the other extreme of the CAI programs are the loose programs such as simulation systems. One such example is a Physics program called Interactive Physics (Knowledge Revolution, 1994). Instead of really doing experiments in laboratories, students can test their ideas by setting up simulations of the experiments in the computer. This not only saves time and trouble, but also allows experiments that are beyond the provisions of school laboratories. For example, students can use the

system to design an experiment testing the collision of two rigid bodies. They can decide on the masses, initial velocities and positions of the two bodies and then the system can simulate the results through animation of the motion of the bodies. There are also tables and graphs showing changes in velocities and other variables. Students could thus have a clear picture of what would happen although the experiment is not actually carried out.

However, a drawback of this kind of program is that apart from the simulations, nothing is done to guide students to discover the principles or laws underlying the experiments. Students are required to explore and devise the hypotheses themselves. Systems like this are termed as “loose systems” since students are not guided in the learning process.

Although currently available educational programs do help students with their quick and accurate responses or interesting and attractive illustrations, the help they offer is either too rigid, in the sense that they do not understand students' errors, or there is no guidance given to the students. This is not to say that they are not useful. It is just that if we are looking for a software to solve homework problems, we may need to develop a new software altogether.

### **Language of Instruction**

One last problem in applying computer technology to school curricula is that most available instructional systems are developed in English. Although English is a medium of instruction in Hong Kong schools, many students may have difficulty in understanding texts written in English. There are some systems written in Chinese imported from Taiwan but not many of these are compatible with the Hong Kong

curricula. Those that can be used on are mostly drill-and-practice systems. On the whole it is not easy to find a computer tutor that can help Hong Kong students in their homework.

### **Requirements of the Personal Tutor**

This review of the current situation as regards available instructional programs shows quite clearly that one way to make individual students to access a personal tutor is to develop a new system. There are two basic requirements of this new system: Firstly, it should be applicable to the curricula of Hong Kong schools, and, hopefully to other parts of the world. Secondly, the system should be able to understand students' responses so as to adjust its tutoring strategies to suit individual needs. In addition, for the system to be used by Chinese students, Chinese, or ideally, a bilingual system should be the instructional medium. Although initially the system may be applicable to a narrow area, it should be capable for future expansion, by being easy modified for other topic areas. It is only when such systems are developed that computer technology can be of help to students in their daily learning.

## **The Present Study -- An Investigation into Electronic Homework**

It is clear that a computer tutor that is compatible with school curricula is highly desirable; but the question is how to bring it about. A computer tutor, at least for the time being, is definitely not to replace a human teacher, but to help the students when live assistance is not available. Asking for a teacher's help in schools should pose no problem to students. Difficulties arise when they need assistance in doing



their homework. This is the ideal starting point for the development of computer tutors.

### **A computer tutor and homework administrator**

In order to help students do homework, computer tutors must, besides providing diagnosing and providing remediation, find some means to order the problems in terms of the degree of difficulty so that students can gain more confidence by doing the easier problems first and then to the difficult ones. Furthermore, the computer tutor could help teachers to understand their students better if it could collect and summarize the errors and weaknesses shown in the students' work. Electronic Homework is designed for such purposes.

Electronic Homework can be composed of two components: the Computer Tutor and the Homework Administrator. The Computer Tutor is an intelligent tutoring system that can provide personal assistance like supplying hints, checking errors, providing remediation and prioritizing problems. The Homework Administrator is intended to be a teacher's assistant in marking the homework and summarizing errors for the teacher's reference. When using Electronic Homework, teachers would simply assign homework by distributing floppy disks containing the assignment for students to do at home. Students can work at their own pace under the guidance of the computer tutor. Next day they return the disks to school. Teachers do not have to mark or correct the homework because it is already done by Electronic Homework. But they can have a clear picture of how the work was done by simply collecting the floppy disks and accessing the summary showing the scores and the errors contained in the piece of

work. This whole process saves teachers a great deal of time which they can now use to improve their teaching.

### **Analysis of the learning process and causes of errors**

Thus Electronic Homework is a convenient tool for teachers besides being an intelligent tutoring system. It is an intelligent tutoring system since it can understand students' errors and provide suitable assistance. It is also a teaching tool since it releases the teacher from several administrative tasks and remediation work. In this sense, it is the first of this kind of computer system. Further, while most previous studies on intelligent tutoring systems focused on simpler domains like multiple column subtraction (Brown & Burton, 1978) and solving first order algebraic equation in one variable (Sleeman, 1987; Moore & Sleeman, 1988; Lewis, Milson & Anderson, 1987), Electronic Homework is aimed at handling a wide range of mathematical problems. Even at the initial stage, the solving of logarithm problems was found to require more complex knowledge than that required by the above systems. Investigation of the knowledge required may thus reveal phenomena that cannot be found in previous studies. Hence, the developing of Electronic Homework is attempted, firstly in the hope that a useful system can be provided for the students and teachers, and secondly that in the process, a more in-depth analysis of learning process and causes of errors could be made.

### **How to Build up Electronic Homework**

Most of the work involved in building up Electronic Homework will be on developing a computer expert that can diagnose students' errors and provide help

when needed. This computer expert cannot be in the form of the traditional computer assisted instructional systems where all possible entries by the students and the corresponding responses are stored. As discussed elsewhere<sup>15</sup>, this will cause problems in that we either cannot know all the possible entries or it is impossible to store all of them. One possible way to do this is to find out all the knowledge required and store them in a system with artificial intelligence techniques. With the knowledge stored and the inference engine provided by artificial intelligence language like Prolog and Lisp, it is possible that the system thus built can generate real time responses to students' entries and thus help them to solve their problems.

### **The Computer Expert**

The question now is finding the knowledge required and storing it. Ways to store knowledge in the system will be discussed in later sections. Knowledge in solving the problems and remedying them is usually obtained from either written materials or interviews of teachers. Knowledge of commonly made errors will be obtained by testing students, interviewing both teachers and students. For some typical errors, computer simulations will be used to investigate the possible underlying causes. All these will be discussed more thoroughly in later chapters. Finally, the computer system also requires the ability to communicate with the users. Such knowledge will also be discussed.

---

<sup>15</sup> For example, page 2.



## **Difficulty of problems**

Besides obtaining and storing knowledge, it may also be required that the system can arrange problems in increasing degree of difficulty so as to motivate students to make progress in learning. To do this, there should be means by which the system can calculate the degree of difficulty of a problem (referred to as problem difficulty from now on) immediately after a problem is entered. As most measures of problem difficulty, like item difficulty level, are calculated after the test items are administered, finding ways to calculate the problem difficulty before a problem is solved by students poses a problem to Electronic Homework. Detailed discussion on ways to overcome this is in Chapter 4 below.

## **Mathematics as the Subject Knowledge**

Electronic Homework was initially designed to help students in mathematics because mathematics knowledge is well structured. Every postulate is carefully hypothesized, every definition is clearly defined and every theorem is logically proved. There is always a logical relation governing any two pieces of mathematics knowledge under the same axiomatic system. Furthermore, in mathematical problems, we can clearly differentiate correct and incorrect solutions whereas in other disciplines, such as appreciation of paintings, it may not be easy to find a commonly accepted standard of good or bad pieces. This may also be the reason why most current intelligent tutoring systems are designed for use in such areas as computer programming and various topics in mathematics. For example, WEST is a computer program for teaching appropriate manipulation of arithmetic expressions (Burton & Brown, 1979, 1982); Leeds Modeling System (LMS) (Sleeman, 1982) is for teaching

algebraic equation solving; Geometry Tutor for teaching geometry (Anderson, Boyle & Yost, 1985) and LISP Tutor (Anderson & Reiser, 1985) for teaching the computer language LISP.

Mathematics is such a large area that it is impossible to include all knowledge in the present study. However, although there may be many different sub-areas in mathematics, there must be a structure common to all. To give a simple example, vectors, matrices and complex numbers are three different areas in mathematics. A vector is a directional quantity like 5 miles toward East. A matrix is an array of numbers like  $\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$  while a complex number is the sum of a real number and an imaginary number like  $3+2i$  where  $i$  is the square root of  $-1$ . Although they have different definitions and look very different, their operations are in many ways alike.

There are many such examples found in mathematics. For the sub-area logarithms used in the present study, the logarithm of a number is in many ways similar to other functions such as sine and cosine. Also, students frequently make similar errors in these sub-areas. For example, a commonly found error in logarithm problems is that students would express  $\log(2+3)$  as  $\log 2 + \log 3$ . In doing sine problems, our experience shows that students would make errors like writing  $\sin(30^\circ+60^\circ)$  as  $\sin 30^\circ + \sin 60^\circ$ . Thus if the principles underlying Electronic Homework is found workable in logarithms, it would be quite possible that they would also work in other mathematics areas. An Electronic Homework system can then be built for all areas in mathematics.

Logarithms thus has its position in mathematics. The fact that the calculators have replaced the using of logarithm table in performing complex calculations cannot



affect logarithms as a representative of a series of functions properties of which should be learned by our students. Evidence further supporting this view comes from the fact that techniques employed in developing the present system enables us to build up knowledge in the form of separate modules. Hence, after developing a system for logarithms, which in fact, consists of knowledge like solving algebraic equations and others, by adding modules of knowledge in handling other areas, the system can be made to tutor as wide an area as possible provided it is within the capability of the hardware.

### **Effect of using Electronic Homework**

Theoretically Electronic Homework is developed for use by all students. Whether it is useful in helping students to do their homework is a subject for investigation. Experience in using the traditional Computer-assisted Instructional programs shows that these systems may not benefit all students (Liu, 1992). It is thus worthwhile to investigate what kind of students may benefit from using intelligent tutoring systems like the Electronic Homework. The system may be made either more suitable for those who benefited less or it could focus only on those who benefited most. The effects of using Electronic Homework will be discussed in later chapters.

### **The Future of Electronic Homework**

At present, Electronic Homework deals only with logarithms, but in the long run, it can be developed into an authoring system that can be a tutor of any mathematics knowledge provided the knowledge can be entered into the system. The



present system thus serves as a prototype showing how human knowledge can be incorporated into a computer system and how computer tutoring can be done.

## Chapter Summary

Although quite a number of computer-assisted instructional systems are now available, few of them are suitable for use in school curricula, either because the systems are not directly related to what our students are learning in school or there is not sufficient guidance given to students in their learning processes. For Hong Kong students in particular, the language of instruction is another problem in using systems designed overseas. This study is thus aimed to design a system to address these difficulties.

It is suggested in the present study that an intelligent tutoring system called Electronic Homework should be designed. By using artificial intelligence techniques, an intelligent tutoring system can store efficiently both the domain-specific knowledge and the pedagogic knowledge and thus can provide a flexible guidance to help students to correct their errors. When compared with the traditional computer-assisted instructional systems, which store all prespecified possible routes and responses, an intelligent tutoring system like Electronic Homework uses less computer space but can handle more complex problems.

However, to design such an intelligent tutoring system requires a thorough understanding of why students make errors and how teachers diagnose and remedy the errors. A major part of this study is thus devoted to the discussion of how and what knowledge is obtained and how it is represented in the system. It is hoped that through

the study of the knowledge required in Electronic Homework that a more in depth analysis of the learning process and the origins of errors can be achieved.

Besides being an effective tutor for the students, Electronic Homework is also intended to be an assistant to teachers in such tasks as scoring and reporting that teachers normally have to do themselves. One additional function that can be carried out by Electronic Homework but is rarely done by human teachers is that, based on the information supplied by the computer tutor, Electronic Homework can give a summary of students' errors. This would be an effective way for teachers to understand their students better and it would provide data for devising remediation.

The effects of using Electronic Homework will have to be investigated before it can be put into real use. The system will be tested by students of varied abilities and attitudes and the results compared with those students who do not use Electronic Homework. In addition, teachers' expert opinion will be collected. Results of all such tests will be the basis for further improvement.

## ***CHAPTER 2***

# **REPRESENTATION OF KNOWLEDGE**

### **Overview**

Before discussing the implementation of Electronic Homework, this chapter first looks at how the required knowledge should be represented in the present system. Although several artificial intelligence methods have been discussed in the preceding chapter<sup>16</sup>, not all of them are suitable for the present system. This chapter starts by arguing that the symbolic approach is more appropriate than the neural network approach in representing knowledge to be used in Electronic Homework. This is then followed by a discussion on the confusion caused by the procedural-declarative knowledge distinction. Finally, it is suggested that an explicit-implicit knowledge distinction should replace the procedural-declarative one, and that the differences in speed and degree of automaticity observed between non-compiled and compiled procedure can be represented by a measure called degree of sophistication. The measure of degree of sophistication can actually provide a finer discrimination

---

<sup>16</sup> Page 14.



between the different states of skill acquisition from completely not acquired to complete acquired.

# How Knowledge is Represented

The development of Electronic Homework requires a wide range of knowledge. The computer tutor, which is one of the two major components of Electronic Homework, requires knowledge to diagnose and remedy students' errors, to arrange problems according to students' standards. Hence the first problem for consideration is how to represent knowledge in the system.

The knowledge required may come from different sources. But before we identify these sources, it may be more appropriate to first discuss how to represent the knowledge. The reason is simple: how we are to store the knowledge will affect how the knowledge is to be obtained.

Currently, two knowledge representation methods are used frequently: the symbolic and the neural network approach. Both approaches try to build models of intelligence but from very different materials (Bechtel & Abrahamsen, 1991).

The symbolic approach is an abstract mathematical representation of human intelligence in computer systems (Wagman, 1993). A system using this approach is called the physical-symbol system which consists of a set of entities called symbols and another set of entities called expressions, each of which is composed of symbols. Also the system contains a collection of processes that operates on expressions to

produce other expressions (Newell and Simon, 1976). Knowledge in this kind of system is thus represented by expressions and the collection of processes. While the expressions are usually explicitly written in the system so that others can read it by using suitable tools, the collection of processes is often implicitly embedded in the system and may not be observed by others. This kind of system is considered as being abstract since each expression is itself representing many instances if the symbols in it are replaced by different values. A rule given in Chapter 1 can be used to show this:

[R19] IF *Number* is divisible by 2

THEN *Number* is an even number.

is actually representing an infinite number of cases when *Number* is instantiated to 2, 4, etc.. Learning in this kind of system is the abstraction process of acquisition of such rules from instances.

In contrast to the abstract view of intelligence, the doctrine of the network approach has as its intellectual heritage the Newtonian world view that construes the mind and mental phenomena as the physical processes of the brain (Wagman, 1993). The notion of intelligence in this approach is that intelligence emerges from the interactions of large numbers of simple processing units (Rumelhart & McClelland, 1986). According to Rumelhart, Hinton, & McClelland (1986), there are eight major aspects of a network model, also called a parallel distributed processing (PDP) model:

- A *set of processing units*
- A *state of activation*
- An *output function* for each unit

- A *pattern of connectivity* among units
- A *propagation rule* for propagating patterns of *activities* through the network of connectivities
- An *activation rule* for combining the inputs impinging on a unit with the current state of that unit to produce a new level of activation for the unit.
- A *learning rule* whereby patterns of connectivity are modified by experience
- An *environment* within which the system must operate

In this kind of system, learning is characterized by the activation of input units and the required output units. Activations then propagate to other units controlled by the propagation rule until an equilibrium state is reached. The knowledge learned is represented by the state of action of the processing units as well as the strength of links that join them. Thus knowledge in this kind of system is represented in terms of basic units and the relationships among them and there is no abstraction process involved.

Thus different methodologies are used to represent knowledge in these two kinds of systems. Since the network model is in a way similar to neurons in the brain while the symbolic approach is the abstraction of human thinking, the former is described as natural, with features inherited from the study of the brain and from the way nature does things. On the other hand, the latter is artificial, related from the beginning to the use of computers (Bechtel & Abrahamsen, 1991).



## Symbolic Expressions or Neural Networks

There has been a lot of debate over the merits of the two approaches. The symbolic approach was claimed to have been tested so extensively over the past 30 years that it can now be regarded as fully established, although over less than the gamut of activities that are called "thinking" (Simon, 1990). Symbolic-processing machines were, however, criticized as failing to provide frameworks for capturing the simple insight into the interactive nature of processing (Rumelhart & McClelland, 1986). On the other hand, the network model was also criticized as the only kind of cognitive processes of which networks seemed capable were those involving associations (Bechtel & Abrahamsen, 1991). There has been a trend to narrow down the gap between the two approaches, but there are still differences. Perhaps an eventual merging of the two would be the best solution. As suggested by Bechtel & Abrahamsen (1991), the key to successful cooperation is that each approach be used for the tasks most suited to it, rather than fighting for the same turf.

There is one way that the present system, and maybe other intelligent tutoring system too, differs from the production system<sup>17</sup> that is intended to simulate human information processing. What is required is not a model to be trained to perform exactly as a human does, nor a precise model of students' behavior. The main concern is to differentiate between correct and incorrect knowledge and the aim of the present system is to reinforce the correct knowledge. Hence, it is the abstraction of the knowledge, i.e., in the form of expressions, that is required. It seems that a network of

---

<sup>17</sup> A set of productions (if-then rules).

interlinked processing units cannot give rise. an easy representation with the current technology Besides, mathematics is itself a discipline that represents knowledge in the form of symbolic expressions. A simple example is that an equation in the form of:

$$x+2=3$$

is actually representing the knowledge of solving problems like how many apples has to be added to 2 to make it 3, or how many persons has be to be added to make a group of three. It is therefore more natural to represent knowledge used in a mathematics tutor in the form of symbolic expressions than its counterpart.

## Procedural and Declarative Knowledge

If expressions are to be used to represent knowledge, the next question will be what knowledge is to be represented. Traditionally, knowledge is divided into two types -- procedural knowledge and declarative knowledge (Anderson, 1990; Gagné, 1985). Declarative knowledge is “factual knowledge that people can report or describe, whereas, procedural knowledge is knowledge people can only manifest in their performance’ (Anderson, 1993). For easy identification of the two types of knowledge, Anderson (1993) also supplied an operational definition in which the types of knowledge are defined in terms of easily identifiable actions. According to his definition, “knowledge that one is able to verbally describe or declare is declarative, while knowledge that can only be inferred from an individual's behavior is considered procedural” (Anderson, 1993). According to Anderson, the knowledge of Washington, DC being the capital of the United States, and memorization of the typewriter keyboard are two pieces of declarative knowledge since one can verbally



state the fact that Washington, DC is the capital of the United States or verbally describe the position of each key on the typewriter keyboard. The typing skill, although it is closely related to memorization of the typewriter keyboard, is a piece of procedural knowledge since you may type very well but you may not be able to describe how your fingers move to the desired positions on the keyboard. It could be the case that when you are asked where the key "l" is you cannot answer until you put your hands on the keyboard and let your fingers lead you to the position. At the same time, there may be two pieces of knowledge coexisting about the same object, one being procedural (moving of arms and fingers to the suitable positions) and the other declarative (memorization of positions of keys).

The operational definition does offer an easy way to differentiate between the two types of knowledge. However, this way of defining knowledge in terms of verbalization is not convincing. Even Anderson himself was not satisfied with this definition. He himself argued against this in that one may not be able to verbally describe the shape of an object but obviously the knowledge about the shape of that object must be declarative since it is a piece of factual knowledge.

Not being satisfied with his own definition Anderson (1993) redefined the two types of knowledge in terms of a theoretical framework. According to Anderson, productions, which are the basic units of procedural knowledge, function by reading information from working memory<sup>18</sup> and writing information to working memory. On the other hand, the information in working memory is declarative knowledge.

---

<sup>18</sup> Human memory is usually divided into two parts: the short-term memory and the long-term memory. Short-term memory only stores information in a very short period of time (approximate 20- 30 seconds) while long-term memory stores information nearly permanently. As thinking is also thought to be occurred in the short-term memory, it is thus also called working memory in some information processing models.



Anderson did not explain why this definition is better than the previous ones in explaining why the declarative knowledge of the shape of an object cannot be verbalized although one can guess that it is because the subject either does not have the procedural knowledge to retrieve the declarative knowledge to the working memory or this subject does not have the procedural knowledge to report the declarative knowledge from working memory. Hence, verbalization is not the necessary condition for knowledge to be declarative although declarative knowledge could always be described if suitable skills (procedural knowledge) are supplied.

### **On Evidence Supporting the Procedural- Declarative Knowledge Distinction**

Anderson tried to provide experimental evidence to support the procedural and declarative distinction. However, it seems that what he provided could only be evidence supporting dichotomy of knowledge but not the procedural and declarative distinction. There may be an infinite number of ways to divide knowledge into two categories, for example, one can categorize knowledge according to whether it is about feeling or whether it is about the human body, but none of this dichotomy of knowledge can be evidence to show that the procedural and declarative distinction does exist. If Anderson is to prove the procedural and declarative distinction according to his definition, he should provide evidence showing that there is knowledge for reading and writing to working memory and there is knowledge that is stored in the working memory. Any other categories can, at most, be partial evidence for the distinction.

According to Anderson (1993), there are six categories of findings that can be evidence for the distinction between declarative and procedural long-term memories<sup>19</sup>. The categories are reportability, associative priming, retrieval asymmetry, acquisition, retention, dissociation. The following sections briefly introduce these categories and argue that these cannot be evidence for the procedural and declarative distinction.

### **Reportable versus Not Reportable**

According to Anderson (1993), procedural knowledge is not reportable but declarative knowledge is potentially reportable. Anderson again does not state clearly what he means by potentially reportable. An educated guess is that declarative knowledge is reportable provided that suitable procedural knowledge is supplied. Then whether or not the knowledge is reportable depends on the supply of suitable of procedural knowledge and not on evidence showing that the two types of knowledge are different. Anderson will need to elaborate if he claims that reportability is an evidence for the existence of declarative and procedural long-term memories.

### **Associative Priming**

Associative priming refers to the effect experienced when one is primed (stimulated) by an item which then makes it faster or easier for those items related to the stimulus to be recalled. For example, when subjects were asked to decide whether a briefly exposed item is true or false, it was found that people recognized that canary is a true word more quickly if they were first shown the word bird rather than first being shown the word rock (Houston, 1986). The reason is that the word canary is

---

<sup>19</sup> One part of human memory system that stores information for a very long period of time.



more closely related to the word bird than the word rock. The word bird is said to be priming the word canary.

According to Anderson (1993), priming can only be found in declarative memory. The example he gives is, when one hears the word computer, there is priming for the word programming (one can read the word programming more rapidly), but not for one's computer programming skills (i.e. one cannot program more rapidly). Because the knowledge of both the words computer and programming is declarative but programming skill is procedural, the example was thus claimed to be evidence for the existence of two long-term memories.

It seems quite unreasonable to try to compare simple tasks like reading a word to complex tasks like programming. Programming is a task that requires knowledge of tens of commands as well as knowledge of how to combine them to form meaningful and executable instructions. This is much more complex than the knowledge of reading one single word. It is possible that priming does occur in both cases, but programming is such a complex procedure that the priming effect within a program may be too small to be observed in the whole process. Hence, even if it is true that the priming effect can only be observed in the so-called declarative memory, it is by no means evidence for the declarative-procedural distinction of knowledge; it is just the complexity of knowledge that counts.

### **Retrieval Asymmetry**

Further evidence claimed by Anderson (1993) to support the declarative and procedural distinction is that there is an asymmetry of access of procedural knowledge that does not exist for declarative knowledge. A rule will work if its conditions match



and will perform its action but it is not possible to have the rule reverse itself and go from action to condition (Anderson, 1993). For example, if there is a rule saying that:

[R21] If it rains, then John will go out for a walk.

Now if it really rains, then we should see that John does go out for a walk otherwise the rule will be false. However, on the other hand, if we now see that John goes out for a walk, we cannot thus say that it is raining because it is possible that John may sometimes go out for a walk when the sun shines.

An example given as evidence for this is what Anderson called principle of specificity. The principle of specificity states that the knowledge acquired through practising one skill cannot be transferred to another skill even if the two skills are related. Experiments done on Lisp programming (McKendree & Anderson, 1987; Pennington & Nicolich, 1991) showed that learning about LISP for coding<sup>20</sup> does not generalize to the use of LISP for purposes of code evaluation<sup>21</sup>.

The coding of Lisp programme is a process from the desired behavior to the code while the evaluation is a process from the code to the desired behavior. The non-transferability of coding knowledge to evaluating knowledge seems to support this retrieval asymmetry of productions. According to Anderson, this is evidence for the existence of such procedural knowledge with its basic unit called productions.

There are two doubtful points about this retrieval asymmetry: first, it seems that even declarative knowledge is asymmetric. For example, we can say that a cat is an animal, but not an animal is a cat. It seems that asymmetry can be found not only in

---

<sup>20</sup> Designing programs for particular purposes.

the so-called procedural knowledge characterized by production-like rules. Also, concerning the principle of specificity, Nicholich and Rahm (1995) argued that the experiments supporting the principle either lacked adequate control or that they did not employ standard measures of transfer. By adding control groups of subjects and analyzing the evaluation and generation tasks in terms of common productions, Nicholich and Rahm (1995) successfully found significant transfer between the tasks which was even more than that predicted by the production common to both tasks. The authors explained that this increased transfer was due to the common declarative knowledge and hence there seemed to be no ground for the asymmetry and specificity of production knowledge.

## Acquisition of Knowledge

On acquisition of knowledge, Anderson claimed that declarative knowledge comes from direct encoding<sup>22</sup> of the environment, whereas procedural knowledge must be compiled from declarative knowledge through practice. Compilation is a word originally used in computer science. Languages used in writing codes for computer programs can be roughly divided into two categories: high-order languages and the low-order languages. High-order languages are those that are similar to human language. For example, a statement written in the computer language Prolog in the form of:

---

<sup>21</sup> Check whether a piece of codes can achieve the desired goal.

<sup>22</sup> As knowledge about an object stored in human brain may not be the same as the real object, direct encoding here simply means the process of translating objects in the environment as codes in the memory.



```
write("abc")
```

would simply mean write the characters "abc" on the screen.

Because of this similarity to human language, they are easy to learn and easy to use. However they have the disadvantage that programs written in high-order languages work very slowly. Examples of these languages are Basic, Pascal, Prolog and FORTRAN.

Low-order languages are those that are similar to language used by computers. The language of the lowest order is called the machine language which is in the form of binary numbers like 11010110. It is easy to imagine that programs written in numbers like this would be very difficult to learn and to use. However, codes in machine language are exact instructions to the computer with 1 meaning on and 0 meaning off. Programs written in machine language can thus work extremely fast. In fact, programs written in high-order languages have to be translated into corresponding machine codes before they can be executed. Translation (called interpretation in computer terms) takes time so programs written in a high-order language thus work much more slowly. When a program written in high-order language is run, every instruction will in turn be translated and then executed. The mechanism responsible for this job is called the interpreter.

As both languages have their advantages and disadvantages, some programmers may like to write their codes first in high-order languages and then covert them into codes in low-order languages before execution. This process of converting the codes into machine codes before execution is called compilation. It is different from interpretation in that one instruction is translated and executed before



the next one is translated and executed, whereas in compilation, all the instructions are translated before execution. Thus compilation and interpretation are exactly the same procedures but executed in a different order.

As compiled codes work much faster than original ones, psychologists borrowed this term to represent those processes that become faster with practice (Anderson, 1990). Thus initially, knowledge is in its declarative form, or high-order language, but with practice, some of the instructions become compiled which then makes the whole process faster. Later, more and more instructions become compiled until no more high-order instructions are left. The knowledge is said to be compiled and has become a piece of procedural knowledge. As it is certain that practice does make a process work faster, compilation does seem to offer a good explanation for the speeding-up of processes.

However, an important point should be noted: Besides the fact that both skills and instructions do work much faster after practice and compilation, there is no other evidence to show that a cognition does exist in the human brain that is doing things similar to a compiler in computers. There is no harm to assume that a human compiler is part of a theory for explaining the skill acquisition process. But it should be remembered that compilation is only needed when there are in fact, two memories. It would be a very serious fault to assume the existence of two types of knowledge in the first place, and then deduce that compilation does exist since practice does make the process faster, but again this knowledge is used as evidence for the existence of two memories in the brain.

### **Retention**

According to Anderson (1993), the retention functions for the two types of memories are independent. When people become more proficient in using procedural knowledge, they sometimes have difficulties in recalling the declarative knowledge that initially gave helpful hints for acquiring the procedural knowledge. Examples of this can be found everywhere: the typing example discussed earlier in this chapter is one, and driving a car is another. When you first learn to drive a car, you may need several verbal hints to guide your operations. But once you have become an expert driver, you will not need any verbal hints. However, if you are asked how you drive a car, you may have to perform the whole process once in order to give a verbal description of your behavior. According to Anderson's definition, the verbal description is declarative knowledge while driving a car is procedural. Hence, acquiring the procedural knowledge may sometimes cause the loss of the declarative knowledge even if the two are closely related.

It should, however, be noted that this observed property does not seem to be restricted to procedural and declarative knowledge. Even for two pieces of closely related declarative knowledge, we may find that the familiarity with one may cause the other one to be forgotten. One simple example is: you have an old friend called Henry whom you have not seen for more than ten years. One year ago, you reared a dog, also called Henry. Now whenever you hear the name Henry, do you immediately think of your old friend or your dog? Naturally it will be your dog. This example, as well as those on typing and driving, can be explained in terms of association. In all the three cases, a stimulus is associated with two alternatives. In the dog example, the name Henry is associated with two objects, your dog and your old friend. In the



driving example, the idea of driving a car is associated with the actual driving and the description of the process. Also, in the typing example, the idea of typing is associated with the actual typing and the layout of the keyboard. Again in all the three examples, one of the alternatives will become more familiar due to practice and the other will recede in the memory. Although in both the driving and the typing examples, one of the alternatives can be claimed to be procedural and the other declarative while in the dog example, both alternatives are declarative. Hence, it should be reasonable to say that this difference in retention is not related to the distinction of procedural and declarative knowledge.

### Dissociation

Anderson (1993) gave examples of amnesiac symptoms to show the dissociation of procedural and declarative memory. Amnesiacs are patients who lose their memory because of brain injuries. Cases reported include HM, who was found unable to recall new events but could acquire new skills (Corkin, 1968) and Korsakoff patients who learned how to read but were unable to recall words they had read (Cohen & Squire, 1980). As both reading and acquiring new skills are procedural knowledge while learning new events and recalling words require declarative knowledge about the events and words, this seems to support the procedural-declarative distinction.

The amnesiac symptoms show clearly that different types of knowledge are stored in different sections of the brain. But why this is related to the procedural-declarative distinction is not clear. Even in the examples cited above, we can see that learning new events and reading words must require descriptive knowledge of the



events and the words. This is considered declarative according to Anderson's definition. However, the recalling of words and learning new events cannot be purely declarative as some knowledge on how to recall or how to learn may be required. Thus we cannot be sure that the inability to recall words or to learn new events is caused by the damage of the declarative memory so that nothing can be stored there, or whether it is because they do not have the necessary skills to store them. More evidence is needed for clarification.

A further comment on this dissociation phenomenon is that: recalling and reading consist of perceptuomotor components while learning new events and recalling words may be purely cognitive. Hence, an alternative explanation for the amnesiacs symptom is that it is due to a perceptuomotor-cognitive distinction rather than procedural-declarative one. Furthermore, we can observe that some of the perceptuomotor skills like hitting a tennis ball can be learned unconsciously. At first, you may miss the ball, but eventually you will be able to successfully hit it for the first time. After practising, the rate of success will become higher and higher. Although this practising may be guided by verbal instructions from the coach or even from yourself, most of the time, it does not seem to be that you are consciously manipulating your muscles, but the technique is improving. Thus it may be true that some perceptuomotor knowledge can be acquired unconsciously, or in other words, without going through the working memory by directly going from the sensory registers to the long-term memory. In this case, the inability to recall new events or to recall words may be due to the fact that the HM and Korsakoff patients were unable to transfer the newly-learned information from the working memory to the long-term memory.

Even Anderson himself would admit that there have been numerous demonstrations of amnesiacs learning skills with strong perceptuomotor components, but relatively few demonstrations of their ability to learn skills that are more purely cognitive (Anderson, 1993, p. 23). Examples found in amnesiacs seem to be more evidence for the perceptuomotor-cognitive distinction rather than the procedural-declarative distinction.

### **Distinction of Knowledge**

In the amnesiac cases, it seems that Anderson has confused the declarative-procedural distinction with the perceptuomotor-cognitive distinction. However, this cannot be evidence against the declarative-procedural distinction. The reason is simple: we can always divide any set of objects into two or even more categories, either for theoretical purposes or based on facts. If the distinction between procedural and declarative knowledge is to be based on facts, then we need concrete evidence to show the existence of the two types of memory. According to this argument, it would seem that currently there is not sufficient evidence for this distinction.

Alternately, if the distinction between declarative and procedural knowledge is for theoretical purposes, i.e., for the easy prediction and explanation of phenomena, there is no problem so long if it can predict accurately and there is no contradiction among its explanations. Two different ways to distinguish knowledge, the perceptuomotor-cognitive and the complex-simple distinction, have been suggested in the above paragraphs. However, this is by no means complete. Any property of knowledge can be used as a criterion for distinction of knowledge so long as it satisfies the above requirement for theoretical distinction of knowledge. The following



paragraphs explore the possibilities of using other knowledge-distinction criterion based on the intention of the procedural-declarative distinction and the practical requirements of computer tutoring systems.

## What is Procedural Knowledge?

The declarative and procedural distinction may become more clear if we can find out exactly what procedural knowledge is. According to Anderson (1990, 1993) and others (e.g., Gagné, 1985), the basic unit of procedural knowledge is described by a production rule. An example of such rules can be found in Anderson's (1993) example production system for addition given below:

[R22] If the goal is to solve an addition problem

and  $c1^{23}$  is the rightmost column without an answer digit

THEN set a subgoal to write out an answer in c1 (Anderson, 1993)

Clearly, knowing that such a rule should be used does imply that this rule will be used. This is perhaps why Anderson and others suggested that this is only a declarative rule and that it will become a piece of procedural knowledge after it has been practised a sufficient number of times. The process of encoding declarative knowledge into procedural knowledge is called compilation. Knowing the rule and applying the rule are thus considered separately as declarative and procedural knowledge. This notion may come from the fact that practising does speed up the process. However, this notion may, on the other hand, overlook some important

<sup>23</sup> There are several columns of numbers in multi-column subtraction and that students are expected to attempt column by column. c1 here refers to the column that a student is currently working on.



features regarding the application of a rule. To correctly apply a rule such as the one shown above, other knowledge besides that explicitly written in it might also be required. For example, one needs the knowledge to "write out an answer in c1". Other examples may be found in the system.

Some of these additional knowledge units may be found in the same system that requires them. However, there are still some that may not be found. For example, in order to solve the addition problem represented by rule [R22], one needs the knowledge to detect that the goal is to solve an addition problem, to detect that c1 is the rightmost column, and to detect whether there is an answer digit. All these elements involve knowledge that is not represented in Anderson's ACT-R system (Anderson, 1993) and other systems like Soar (Laird, Newell & Rosenbloom, 1987; Newell, 1991).

It must be acknowledged that no system can include all knowledge and that all users are assumed to possess some knowledge before using the system. Researchers (Anderson, 1993; Lewis, Milson, & Anderson, 1987) suggested some knowledge grains<sup>24</sup> have to be assumed so that no further dissection on these grains will be done although there are possibilities. Grains are different from basic units of knowledge such as productions and propositions but are composed of them. Knowledge grains are not explicitly expressed in the cognitive system but it is assumed that everyone who uses this system should already have the knowledge represented by these grains.

---

<sup>24</sup> Assumed smallest units of knowledge in the cognitive system.

## Explicit versus Implicit Knowledge

Grains are some kind of implicit knowledge that are not explicitly represented in any system. There are some other knowledge units that can be treated as implicit knowledge. The following rule explicitly states a knowledge piece that is required to solve a multi-column subtraction problem:

[R22] If        the goal is to solve an addition problem

                 and  $c1^{25}$  is the rightmost column without an answer digit

THEN set a subgoal to write out an answer in  $c1$                       (Anderson, 1993)

However, it seems that in Anderson's system and other systems like Soar (Laird, Newell & Rosenbloom, 1987; Newell, 1991), once such a rule is incorporated into the system, it will automatically be selected and applied whenever appropriate. The knowledge of selecting and applying rules is not implicit from the point of view of the designer of the computer language which is used to develop this kind of system. For the persons who designed the cognitive system, they just have to be careful when designing the rules but they do not have to worry about how these rules are applied. To these designers, the knowledge of selecting and applying knowledge is implicit that this knowledge is assumed.

Implicit knowledge is in some ways like procedural knowledge in the sense that they are both non-reportable. However, implicit knowledge is neither knowledge of how to do things, nor knowledge of reading and writing to working memory

---

<sup>25</sup> There are several columns of numbers in multi-column subtraction and that students are expected to attempt column by column.  $c1$  here refers to the column that a student is currently working on.

(Anderson, 1993). Knowledge such as knowing that c1 is the rightmost column in rule [R22] above should be declarative knowledge according to Anderson's categorization, but it is implicit since there is no part in the system stating what the rightmost column and what is a column is.

Hence, any system would consist of two types of knowledge, explicit and implicit. Explicit knowledge is the knowledge that is directly applied when solving problems that the system is designed for. Implicit knowledge is all other knowledge that is required. The use of explicitness as criterion for categorization solves confusion caused by the procedural-declarative distinction. However, this categorization of knowledge is only for the convenience of designing computer systems, there has been no claim that there should be two memories in the brain, each corresponding to one knowledge stated here. Although some of the evidence claimed to be supporting the procedural-declarative distinction can also be applied to this implicit-explicit distinction, it will not be dealt with here since it is not the main concern of the present study.

## **Degree of Sophistication versus Proceduralization**

Anderson (1983, 1987, 1990, 1993) and others (Gagné, 1985) argue that the acquisition of skills has to be done through a process of knowledge compilation. According to Anderson, skill involved in doing a certain task initially consists of several pieces of declarative knowledge. Through practice, each declarative knowledge piece will be compiled into corresponding procedural knowledge pieces.



## Representation of Knowledge

Pieces of procedural knowledge are represented by productions. When several productions are placed in the working memory at the same time, they will be combined into one production. The processes of changing declarative knowledge into procedural ones and combining several productions are called proceduralization and composition respectively while the whole process is called knowledge compilation. An example given by Gagné (1985) serves to illustrate this process:

Suppose now the task is to add fractions. Initially, the students would have a list of steps either from the textbook or from the teacher's instructions. The list may be like the steps shown in Table 2.1:

Table 2.1

The Steps Involved in Adding Fractions (taken from Gagné, 1985).

- 
1. Find the least common denominator.
  2. Divide the denominator of the first fraction into the least common denominator.
  3. Multiply the result of step 2 by the numerator of the first fraction.
  4. Write the result of step 3 above a line and the least common denominator below that line.
  5. Repeat step 2-4 for the second fraction.
  6. Add the numerators of the two fractions written down in step 4.
  7. Write the result of step 6 as a numerator.
  8. Write the least common denominator as the denominator.
  9. If the numerator and denominator have a common factor, divide them by this factor and write the result.
-

When these steps are read into the brain, they are in the form of declarative knowledge. After that, one translates each step into productions through practising individual steps. Table 2.2 shows productions for the first three steps in the procedure:

When these productions are formed and are used together for sufficient number of times, all the proceduralized productions will be combined as one large production as shown in Table 2.3.



Table 2.2

Production Representations For the First Three Steps in Adding Fractions

---

P1	IF	My GOAL is to add FRACTIONS and there are two FRACTIONS to add
	THEN	Set SUBGOAL to find LEAST COMMON DENOMINATOR.
P2	IF	My GOAL is to add FRACTIONS and there are two FRACTIONS to add and LEAST COMMON DENOMINATOR is known
	THEN	Divide DENOMINATOR of FRACTION 1 into LEAST COMMON DENOMINATOR to get RESULT 1.
P3	IF	My GOAL is to add FRACTIONS and there are two FRACTIONS to add and there are two FRACTIONS to add and I have RESULT 1
	THEN	Multiply NUMERATOR of FRACTION 1 by RESULT 1.

---

Table 2.3

A Production for Computing the Least Common Denominator.

IF	GOAL is to find LEAST COMMON DENOMINATOR
	and there is more than one FRACTION
THEN	Multiply all DENOMINATORS to get PRODUCT
	Identify COMMON FACTORS among DENOMINATORS and PRODUCT
	Divide PRODUCT by COMMON FACTORS

The composed production works faster and requires less working memory space since there is only one production to be taken care of.

The skills acquired through compilation and proceduralization are thus characterized by its automaticity (i.e. the skill will be applied automatically whenever the required conditions are satisfied) and faster execution (the job will be done faster than originally when only the declarative knowledge is provided).

Since it is true that practice speeds up performance, knowledge compilation does offer a reasonable explanation for this phenomenon if there is really a procedural-declarative distinction. However, if there is, in fact, no such thing as procedural-declarative distinction, the concept of proceduralization will have to be either changed or abandoned since the translation from one knowledge to another will

be meaningless. In addition, the idea of proceduralization will also have to be modified since there is no more compiled knowledge.

According to an earlier argument, besides the fact that if we practise a skill several times, we can actually perform the skill faster, there is no evidence for the existence of a cognition that does the same work as a compiler in a computer language system. Further, in Anderson's skill acquisition theory (1983), subskills are acquired through proceduralization while skills are acquired through composition of subskills. It is strange that two different processes are required for the acquisition of skills and subskills as subskills are themselves skills. For example, to type a passage is a skill that requires the subskills to type characters, which in turn are skills that require the subskills to move the fingers and press keys. It would be very hard to say which of the skills are acquired through composition and which through proceduralization.

On the other hand, if the term subskill is used in a relative sense, i.e. a skill can be a subskill of other skills but may also consist of other subskills, then the acquisition of skills can be explained solely in terms of composition of subskills which can then be explained in terms of their subskills. If this is the case, then there will be no need for proceduralization and even no need for the procedural-declarative distinction. Instead, knowledge can be described in terms of its degree of sophistication: knowledge pieces that are formed by composing other pieces can be thought as of higher degree than its components, while the components can be thought of as being of lower degree. Hence the grains<sup>26</sup> can be thought of as the lowest degree of sophistication, knowledge made of these grains through composition are of higher degree and so on. For the same skill,

---

<sup>26</sup> Assumed smallest units of knowledge in the cognitive system, described earlier in this chapter.



there may be more than one knowledge piece to accomplish it. These knowledge pieces may actually be made of the same set of knowledge grains but only differ by the degree of sophistication of their components. Some of these knowledge pieces are composed of large units formed by several cycles of composition. While some of the others are made of less composed or even uncomposed units. These knowledge pieces can also be thought of as having different degrees of sophistication: those with more composed units higher and the others lower. Of course, a detailed calculation procedure must be worked for a more accurate picture. However, if we recall our experience in learning to drive a car or any other skills, there seems to be some intermediate stages between the initial stage, during which we have to memorize all the verbal instructions, and the final stage when driving becomes automatic and effortless. The procedural-declarative distinction seems to be claiming that there could only be the initial and the final stages. In the present degree of sophistication description, many more stages are accepted. This latter notion seems to be closer to the real situation.

## Notation of Knowledge

A lot of confusion between procedural and declarative knowledge may be due to the fact that different notations are used to represent facts and actions. In most production systems, facts are usually represented in statements like:

## Representation of Knowledge

[R23]            `isa(cat,animal)`<sup>27</sup>-- A cat is a animal .

[R24]            `color(rose,red)` -- The color of rose is red.

On the other hand, actions are represented as if-then rules. For example, to find the sum of two numbers a and b, we would need a production:

[R25]            if sum of a and b is required then calculate a+b and report

To sort two numbers a and b, the following rule may be required:

[R26]            if  $a > b$  then put a before b .

That may be why theorists in artificial intelligence referred to knowledge about facts as declarative knowledge and knowledge about how to act as procedural knowledge. According to Chabris (1991), declarative knowledge is the knowledge of "knowing what" and it describes the information necessary to solve the problem in a general way, without providing a direct method for its solution. On the other hand, procedural knowledge is the knowledge of "knowing how", it encodes knowledge as process, asserting that a procedure for solving a problem is the knowledge of how to solve it. Examples given by Chabris can illustrate this more clearly. In his words:

---

<sup>27</sup> Computers do not use natural language. To input knowledge into a computer system, the knowledge has to be written in computer syntax. This is an example of the syntax used in the computer language Prolog.

*The knowledge to solve a mechanics problem in physics could be represented declaratively as a set of differential equations. These represent all the necessary information, and imply that the value(s) which satisfy all the constraints simultaneously are the solutions. A procedural representation might specify a series of steps the problem solver should take, one after the other, to arrive at the solution from the initial description of the problem (Chabris, 1991. page 30).*

In terms of Chabris's definitions, both types of knowledge can be represented as descriptions: declarative knowledge is the description of what could be used to solve the problem while procedural knowledge is the description of what should be done in order to solve the problem. The difference between the two types of knowledge may just be in their ways of organizing the components but not in the ways they are stored in the memory system.

However, according to Anderson, procedural knowledge is not reportable, hence rules [R25] and [R26] can only be declarative knowledge since they are the descriptions of what actions should be done under the conditions specified in the condition parts. It is only through proceduralization that these two declarative knowledge pieces can become procedural knowledge which is characterized by being non-reportable. But again, these two newly compiled pieces are represented by the rules [R25] and [R26]. This dual-representation causes much confusion that should be avoided.

It should be pointed out that this is not rare in natural language. An example based on that given by Leung (1967) shows that:



Hong Kong is a small island.

Hong Kong is a bi-syllabic word.

In the first sentence, the words Hong Kong are used to represent the place Hong Kong, while in the second, it is used to represent the name of Hong Kong, but they appear to be identical in the two sentences. Other similar sentences can be easily found in many natural languages we use.

### What Should Be Done But Not What Is Actually Done

The fact that dual-representation is so commonly used does not imply that this convention should be retained. People may be so used to this kind of convention that they would accept the dual-representation of productions without question. It is, however, better to avoid it so as to reduce the confusion that it would cause. In previous sections, it is argued that there is not enough evidence for the procedural-declarative distinction. Further, if the explicit and implicit distinction is used, there is no more need for the procedural-declarative distinction. If there is only one memory in the brain, this dual-representation confusion will automatically disappear. As the rules [R25] and [R26] are explicitly represented in the system, they are explicit knowledge that describe what actions should be done with knowledge implicit in the system responsible for the execution. Knowledge explicitly stored in production systems is thus a description of what actions should be done under specified conditions and not the actual actions. There is thus no more confusion caused by dual-representation of productions.

## Chapter Summary

This chapter describes how knowledge should be represented in the present system. A major portion has been devoted to the discussion of the procedural-declarative distinction. Although it is argued here that evidence claiming to be supporting the procedural-declarative distinction is irrelevant, it is by no means saying that there is only one memory system in the brain. In fact, there is evidence to show that different portions of the brain have different functions (Kalat, 1984) but neither of these distinct memories can be found to have declarative or procedural properties. As no evidence can be found to the contrary, there is still the possibility that the procedural-declarative distinction is true. But of course we will have to await further knowledge about the brain in order to verify this.

It is, of course, possible to assume theoretical distinction of knowledge if it helps to explain the complex processes involved in solving problems. However, this procedural-declarative distinction has a serious drawback making it impossible to be adopted into the present system. As procedural knowledge is traditionally represented by production which again represents the corresponding uncompiled declarative knowledge, a production thus has a dual-representation. It sometimes represents the declarative knowledge to be compiled but sometimes represents the compiled knowledge. This dual-representation not only causes confusion, it also makes it difficult to be represented in any production systems.

Instead of the procedural-declarative distinction, the explicit-implicit distinction of knowledge is proposed here. This explicit-implicit distinction is only a

theoretical construct; there is no experimental evidence to show that this distinction does exist. The use of this distinction is just for convenience since in any knowledge system, there is always knowledge not explicitly specified but is implicitly assumed. Examples of implicit knowledge are basic knowledge units called grains and inference rules. For explicit knowledge, it includes all the knowledge pieces written in the system.

A major function of the production-declarative distinction is to explain the skill acquisition process. According to Anderson (1983, 1987, 1993), skill is procedural knowledge and is acquired through knowledge compilation. Knowledge compilation is done through two processes: proceduralization and composition. As there is no more procedural-declarative distinction, proceduralization will no longer be needed. Composition is the sole cause for skill acquisition. Composition is the process of combining several productions into one and the execution of one production will be faster than the execution of several. Hence even if there is no more proceduralization, composed production still works faster than that not composed. In terms of composition only, a skill acquisition process can be described as this: initially, a skill consists of several component skills, each of them working separately. After practice, some of the consecutive component skills compose to become groups. The skill then works faster since it now involves the execution of fewer productions. After further practice, more and more component skills are composed as groups, also separate groups and groups may be composed as larger groups. The skill now works even faster until at last all components are composed as one production.



Composition of productions can be described in terms of a parameter called degree of sophistication. Composed production are thought of as of a higher degree while the components are of a lower degree. Thus in the above skill acquisition description, the initial state of the skill has the lowest degree of sophistication since none of the components is composed while the final state of skill is of the highest. Skill acquisition can now be described in terms of the increasing degree of sophistication. There may be many intermediate states between the two extremes. When we compare this to either compiled or uncompiled versions used in the procedural and declarative memory systems, it would be quite clear that the former is closer to the real situation since we all know that skills are acquired continuously rather than rising suddenly from nothing to full mastery. Applying the lowest degree knowledge with inference rules is like using some general methods, while applying the highly composed knowledge can be thought of as using some domain-specific methods. Winograd (1985) suggested that an entire hierarchy of methods should be attached at all levels of the generalization hierarchy of the concepts in the problem domain and there should be no sharp division between specific and general methods. The use of degrees of sophistication, is in some ways agrees with this notion.

It is argued above that the procedural-declarative distinction may, in a way, help to explain the skill acquisition process if it is treated only as a theoretical distinction. It is just the dual-representation problem that renders the distinction unsuitable for use. However, there is nothing wrong with the knowing what and knowing how distinction. In our common use, as well as knowledge represented in artificial intelligence languages, it is common to find that some knowledge is better expressed in the form of "what is" and others in the form of "how to". Examples of the

former are the knowledge shown in [R23] and [R24] above, while the rules [R25] and [R26] represent the latter. But it should be borne in mind that rules such as [R25] and [R26] are only descriptions of how to do an action but not the exact action. There should be no compiled or uncompiled versions. By just abandoning the notion of proceduralization, the distinction of procedural rules and factual rules will be a convenient tool to describe knowledge.

## ***CHAPTER 3***

# **WHAT KNOWLEDGE TO INCORPORATE AND HOW**

### **Overview**

Chapter 2 discussed the representation of knowledge. This chapter discusses what kind of knowledge is to be incorporated and how this can be done. Four types of knowledge, namely, the domain knowledge, the student model, tutoring knowledge and communication knowledge, are identified and incorporated into four different modules. Figure 3.1 depicts what these modules are and the topics discussed under each of them.



Knowledge Represented in Electronic Homework

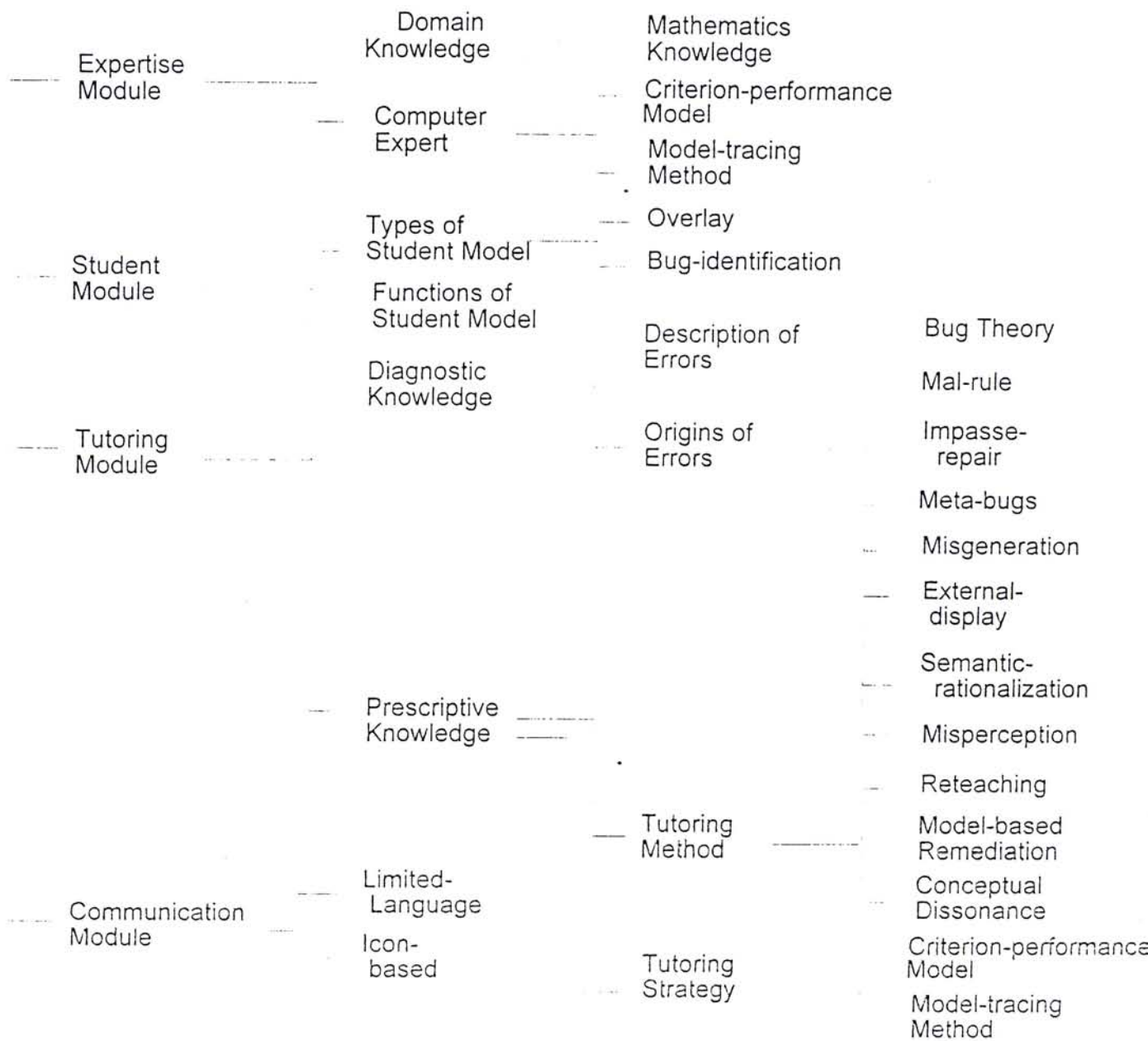


Figure 3.1 An Overview of the structure of Topics in Chapter 3.

A theoretical background of how knowledge should be represented was given in previous chapters: Knowledge will be represented as explicit rules while some basic knowledge and inference rules will be assumed implicitly. The question that follows is what knowledge is to be incorporated. To be a tutor in mathematics, inevitably two types of knowledge should be included: mathematics knowledge and knowledge of the teaching of mathematics. The knowledge on how to teach mathematics may further be divided into knowledge on how to teach effectively, how to motivate and how to respond to students' answers. The system may thus have to maintain a catalog of students' errors and ways to remove them in order to respond suitably to students' answers and queries. All this knowledge has to be communicated to students in effective ways, hence it would be necessary to incorporate knowledge on how to communicate between the computer system and the user. A computer tutor thus requires different types of knowledge and each of these will be discussed in detail in this chapter.

## **Separate Storage for Different Types of Knowledge**

Before the different types of knowledge can be discussed, the problem of how to store them in a computer system has to be solved since a proper storing method would not only enable efficient use of this knowledge in the future, but would also facilitate future expansion. To enable future expansion, the system should be so designed that adding new knowledge would not require major restructuring of the old knowledge. In traditional computer-assisted instructional systems, decision points are

pre-specified and responses given to students are linked to particular decision points. Thus each tutoring or diagnosing strategy is specific to the decision point to which it is linked and it is not easy to adapt it for other decision points. Such a knowledge-storing method is duplicated and memory wasted. Moreover, when the system has to solve new problems, adding new decision points will require a restructuring of the system and each knowledge piece has to be newly incorporated even though some of them can be found embedded somewhere in the system. Thus it is not easy to expand such a system.

The designing principles of traditional computer-assisted instructional systems limit the number of ways its knowledge is stored. For current intelligent tutoring systems, with the help of artificial intelligence techniques, knowledge can be separately represented as rules which allow the system to make decisions and to reason during the teaching interaction (Goodyear, 1991). New knowledge can be added to the system by simply adding new rules when needed since old knowledge pieces can be reused.

## Different Types of Knowledge

It is commonly agreed that four sets of knowledge should be included though they may be named differently (Yazdani, 1987; Woolf, 1987; Park, 1991). The sets are:

- Domain knowledge - on the subject domain; in this case, it is mathematics knowledge.



- Student model - knowledge of the student as perceived by the computer system and includes such knowledge as misconception and errors made by the student.
- Tutorial knowledge - on how to teach the students.
- Communication knowledge - on how to communicate with the learner through the computer.

Each set of knowledge refers to a different kind of knowledge that an intelligent tutoring system (referred to as ITS from now on) should have, though not all ITSs would incorporate all of them. Also, the positions of each set of knowledge in a system may not be identical. Earlier systems may mix all kinds of knowledge together, while later systems may put them into separate modules. Hence a typical ITS would have four modules - the expert module, the student module, the tutoring module and the communication module, each containing the domain knowledge, the student model, the tutoring knowledge and communication knowledge respectively. The following sections discuss what knowledge should be incorporated in these four modules.

### The Expert module

This module contains the knowledge that the system is imparting to the student. It is called an expert module since it includes what an expert in the subject area concerned should know. According to Roberts & Park (1991), there are two aspects to this module - the domain knowledge base and the criterion-performance model. The domain knowledge base includes both the knowledge of the contents to be

taught and the knowledge on how to use the content knowledge to solve related problems. The criterion-performance model is a computer-based expert that solves the same problem given to the student so that the system can evaluate the student's performance.

### Domain Knowledge Base

As the present system is intended to teach logarithmic knowledge, it thus includes knowledge required to solve logarithm problems. However, solving logarithm problems may require other mathematics knowledge like solving algebraic equations, simplifying algebraic expressions, factorizing numbers or algebraic expressions. The knowledge base might therefore include a large number of rules. According to Lewis, Milson, & Anderson (1987), there are two types of such rules—the strategic and the axiomatic rules. Strategic rules state what strategies would be used whenever certain patterns are observed, while axiomatic rules correspond to behaviors according to mathematics axioms. The following examples found in The Teacher's Apprentice (Lewis, Milson, & Anderson, 1987) serve to illustrate this difference:

[R31] IF the equation to be solved contains a subexpression of the form  
 $num(term1 + term2)$

THEN set as a subgoal to distribute *num* over *term1* and *term2*

[R32] IF the goal is to distribute *num* over *term1* and *term2*

THEN set the subgoal to multiply *num* times *term1*

- AND        set the subgoal to multiply *num* times *term2*
- AND        set the subgoal to combine the previous results with +
- [R33]    IF        the goal is to multiply *num* times *term*
- THEN     write the product of *num* and *term*
- [R34]    IF        the goal is to combine *term1* and *term2* with a +
- THEN     write *term1* + *term2*

(Words in italics are variables.)

In the above examples, the rule [R31] recognizes that distribution is applicable to the equation and sets the subgoal to distribute *num* over *term1* and *term2*. It is a strategic rule. The other three are axiomatic rules since they show the actions according to distributive law, multiplication and addition facts respectively.

The advantage of separating strategic rules from axiomatic rules is that the tutor's cognitive load<sup>28</sup> can be lightened so that he or she can focus separately on the student's strategic decisions at some points and application of axiomatic knowledge at others (Lewis, Milson & Anderson, 1987). Also, according to Lewis, Milson & Anderson (1987), simply learning how an axiom manipulates symbols may be easier than learning when to apply that axiom in service of problem solving. However, both strategic and axiomatic components of a skill must both be well learned if the skill is to be applied successfully in problem-solving. Both types of rules are to be incorporated.

---

<sup>28</sup> A hypothetical term. When people put many things in the mind, they are said to be having a heavy cognitive load.



## **Criterion-performance model Versus Model-tracing**

There has been some argument on whether the criterion-performance model is required for a computer tutor. Although most systems would have the knowledge to work out a correct problem-solving path as a criterion, called the criterion-performance model, for the students' process, not all of them would like to do so. In some systems, like PIXIE (Sleeman, 1987; Moore & Sleeman, 1988), the solutions to a problem, whether correct or incorrect, are generated before the problem is presented to the students. Students' answers are then compared with these generated solutions as models and instructions will be given to those answers found identical to an incorrect model. This method of checking students' answers is called the criterion-performance model method. An example taken from PIXIE is given in Figure 3.2 to illustrate how this kind of system works.

In Figure 3.2, initially the student is given a problem to solve (lines 1 to 2) and after performing one step (line 3) claims that the problem is finished (line 4). The computer then asks the student to further simplify and these are shown from line 5 to line 7. After the problem is finished, the computer checks the student's answer by comparing it with the catalog of models stored in the computer system and decides on one of models. This model is then presented to the student for verification (lines 12 to 18). When this model is confirmed by the student, the computer explains why the error was made and how to correct it (lines 21 to 22). The correct procedure to solve the problem is then given to help the student to correct any misconception he or she might have. Hence the whole tutoring process is characterized by comparing the of student's answer to the pre-generated models. Instructions to correct the errors are given based on these models.

### An Example of Criterion-performance Model.

- 
- 1.Solve (f=Finished, q=Quit early)
  2. $7x = 2*2$
  3. $7x = 4$
  - 4.f
  
  - 5.Simplify further  $7x = 4$
  6. $x = 4$
  - 7.f
  
  - 8.Thank you,
  - 9.but you didn't get that one right.
  - 10.We will look at that task again.
  
  - 11.Here we are considering the task  $7x = 2*2$
  
  - 12.Did you work the task something like this?
  - 13.
  14. $7x = 2*2$
  15. $7x = 4$
  16. $7x/7 = 4$
  17. $x = 4$
  - 18.Please type YES or NO: y
  
  - 19.it seems you changed  $7x = 4$  to something like  $7x/7 = 4$
  
  - 20.This is WRONG.
  - 21.DO NOT only divide ONE side by the coefficient.
  - 22.You should divide both sides of the equation by the coefficient to change  $7x = 4$  to  $7x/7 = 4/7$ .
  
  - 23.Now let me show you how you SHOULD have worked the whole task.
  

24. $7x = 2*2$	multiply 2 numbers
25. $7x = 4$	divide both sides of the equation by the coefficient
26. $7x/7 = 4/7$	divide
27. $x = 4/7$	Finished.

---

Figure 3.2 An Example of Student Interaction in using PIXIE

**An Example of Model-tracing Method**

In systems like The Teacher's Apprentice (Reiser, Anderson, & Farrell, 1985) and LISPITS (Corbett & Anderson, 1992), there is no specific criterion-performance model. Instead, they use a "model tracing" method of tutoring. At each state (step) of the process, the system infers the learner's internal state by matching his output with the problem state generated by using ideal and incorrect rules (referred to as buggy rules). Instructions will be given according to this inference. An example taken from LISPITS (Corbett & Anderson, 1992) is shown in Figure 3.3 to illustrate how this kind of system works.

(a)

<b>Tutor Window</b>
Define a function called pal that takes a single list as an argument and returns a palindro that is twice as long. A palindrome is a list that reads the same forward and backward. F example, (pal '(a b c)) returns (a b c c b a)
<b>Code Window</b>
(defun <name> <parameters> <body>)

(b)

<b>Tutor Window</b>
Remember that you are trying to get the parameter list of the function here. You should n be calling the function list. If you were thinking of using list as a parameter name, it is a b idea because you might get confused between the function and the parameter.
<b>Code Window</b>
(defun pal (list) <process>))

Figure 3.3 An Example of Student-Computer Interaction in LISPITS



LISPITS is an intelligent tutoring system on the computer language LISP designed to be used by college students. Both Figure 3.3 (a) and (b) show the terminal screen when students are working on an exercise. In both cases, the screen is divided into two, with a tutor window at the top and a code window at the bottom. The tutor communicates with the student by means of the tutor window; the problem description appears in this window at the beginning of an exercise and remains there except when the student makes a mistake. The code the student types appears in the code window.

Figure 3.3 (a) shows the initial stage of the interaction during which the problem description is displayed and the student has just typed in a left parenthesis and `defun`, and LISPITS has responded by putting up a template for the student: a matched right parenthesis and three goal symbols are displayed on the screen in angle-brackets. LISPITS also highlighted the goal symbol which the student must work on next.

Figure 3.3 (b) shows the stage when the student has just entered a correct name but an incorrect parameter. LISPITS identifies this error and displays the prescription in the Tutor Window. Exactly what the error is and how the computer diagnoses the error will not be discussed here because many technical details on the use of the language LISP are involved. However, this example shows how LISPITS works by checking each step the student works and how the student's error is corrected.

Experimental results showed that students using LISPITS complete the coding exercises substantially more rapidly than those working on their own, although not as fast as students working with a human tutor. Both the model-tracing and the

criterion-performance model have their advantages and disadvantages. The criterion-performance model approach might require a lot of space to store the models and also a long period of time to generate all the possible models while the model-tracing approach might prevent the student from learning by making errors. Detailed discussion on whether the criterion-performance model or model tracing method should be employed in the present study is presented in later sections on tutoring strategies.

## **The Student Module**

If the goal of an intelligent tutoring system is individualized instruction, it would be better for the system to keep a catalog of possible errors so that immediate assistance could be given to students making different errors. As it is normally impossible to have students directly put in their knowledge, it is necessary for the system to infer the knowledge from students' responses. Knowledge inferred about the students is thus called the student model in the sense that it may not be exactly what was in the students' mind.

Regarding what knowledge about students should be incorporated into a computer tutor system, Self (1988) suggested that a student module should enable the tutoring system to answer questions about the student using the system. Questions are broadly of four types: What can the student do? What does the student know about? What type of student is he or she? and What has the student done? All these may have to be answered by acquiring different types of knowledge. In addition, Cumming & Self (1991) further raised the following three questions to be answered by intelligent tutoring system designers:

- What should be modeled?
- What detail is necessary?
- How closely tailored to individual learners does the model need to be?

The first question seems to ask the same as the four types of questions in the last paragraph. However, they are not exactly identical since knowledge in an Intelligent Tutoring System is not necessarily to be modeled, since some may be inferred. Hence, besides considering what knowledge should be incorporated in an intelligent tutoring system, we have to also consider how this knowledge should be incorporated.

Self (1988) further suggested that a student model should be a 4-tuple  $\langle P, C, T, H \rangle$ , where  $P$  describes procedural knowledge,  $C$  conceptual knowledge,  $T$  individual traits and  $H$  the history.  $T$  is typically a set of labels, e.g. introvert, blind, bored, etc. describing the student.  $H$  may be a transcript of the interactive session, summarized and interpreted to describe significant events.

### Types of student models

In order that the above questions and suggestions can be handled, student models have to be developed in an intelligent tutoring system. Traditionally, student modeling is in two broad categories: the quantitative method (Park & Seidel, 1991), which is mostly used in conventional computer-based instruction (CBI) and is thus not elaborated further here, and the qualitative method. Clancey (1988), in defining qualitative models, says “The qualitative model is neither numeric nor physical analogues. Rather, it describes objects and processes in terms of spatial, temporal, and

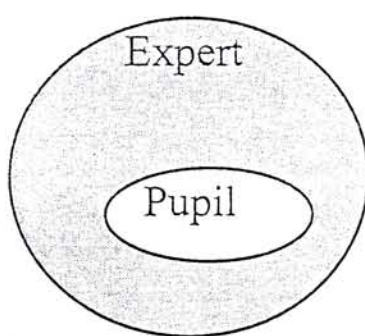


causal relations.” There have been mainly two types of methods used to model the students:

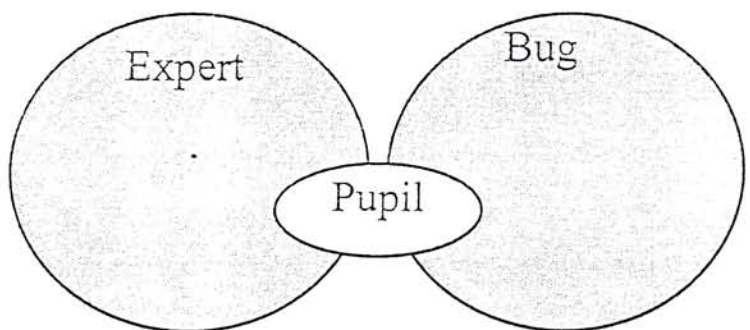
1. Overlay model: the student's performance is compared to that of the computer expert. The expert's competence is assumed to be broken into a set of skills so small that the pupil either has them or doesn't (Elsom-cook, 1988). In other words, the student has part of the expert's knowledge.

2. Bug identification method: the student model contains both domain knowledge as rules and misconceptions and errors (bugs) as variants of rules. In this case, the student model includes something that the expert does not have. It is thought to be more realistic than the first type.

The following shows the relationship of the student model with respect to the expert's behavior and the bugs (Elsom-cook, 1988).



Overlay Model



Bug-identification Model

While the present system is designed to help students to correct their errors and because the overlay model does not contain such knowledge, it would be reasonable to say that the bug-identification model would be more appropriate for the present use.

## Functions of Student Models

Self (1988) identified six functions of a student model as follows:

1. corrective: to help eradicate bugs (errors) in the student's model;
2. elaborative: to help extend what is described in the system (which may be considered 'correct' but 'incomplete')
3. strategic: to help initiate more significant changes in the tutorial strategy than the tactical decisions of (1) and (2) above.
4. diagnostic: to help resolve the contents in the student model.
5. predictive: to help determine the student's likely response to tutorial actions.
6. evaluative: to help assess the student or ICAI<sup>29</sup> system.

To achieve these functions, the student model would have to be precisely constructed so that a student's individual needs, learning styles and previous knowledge can be represented. However, there are factors that impede precise student modeling. According to Self (1990), these factors are:

1. Combinatorial explosion. If according to PIXIE (Sleeman, 1987; Moore & Sleeman, 1988), a catalogue is kept of the paths used by the student to solve the problem, then the number of possible combinations will be too large to be incorporated into a common computer system when more knowledge is involved in a system.

---

<sup>29</sup> Intelligent computer-assisted instruction, refer to page 17 for more detailed description.

2. Lack of global view. The combinatorial problem can be solved by constraining the student to the smallest and analyzable step, with the consequent imposition of a rigid tutorial. The Teacher's Apprentice (Reiser, Anderson, & Farrell, 1985) described earlier on expert module section is such a system. However, this will result in a situation where it would be difficult to have an overall understanding of the student since only a step at a time is checked.
3. Students' prior knowledge. Students' decision-making ability may depend on a lot of prior knowledge which may not be directly related to the learning materials. For example, Shrager (1987) showed that people learned by experimentation to operate a programmable toy by drawing analogies with other programmable devices such as clocks. The use of such commonsense knowledge is usual but it is difficult to identify, not to mention incorporate into an Intelligent Tutoring System.
4. Immediate learning context. Kolodner (1983) showed that students attempting physics problem drew surface analogies with immediately preceding problems. This means the Intelligent Tutoring System needs to maintain an episodic memory in order to better provoke productive analogies and to understand the source of mistaken analogies (Self, 1988).
5. Personal learning preferences, styles and strategies. Ideally, these should also be represented with the student model for better individualized instruction.

The constraints mentioned above make the construction of precise student models almost impossible. However, Self (1988) argued that it is not essential for Intelligent Tutoring Systems possess precise student models. The modeling problem



is practically attainable if we back off from the grand vision and adopt more realistic aims. He described four slogans for achieving these:

1. "Avoid guessing -- get the student to tell you what you need to know". Self suggested that rather than attempting to develop better ways of inferring missing steps, it is better to design interactions through which the information needed for building student models is provided non-intrusively as an intrinsic part of problem solving. A possible way of doing this is by providing alternative ways for students to choose.
2. "Don't diagnose what you can't treat". Some people would say that many factors such as cognitive styles, motivation and personality characteristics should be added to the student model. Self (1988) thought that this would achieve nothing. He further claimed that the student model and tutoring procedure should be developed in tandem, not separately. Any proposed feature of a student model should be explicitly linked with existing educational evidence which justifies it.
3. "Empathize with the student's beliefs, don't label them as bugs<sup>30</sup>". The traditional mal-rule approach has been questioned for its usefulness as representations of students' errors because of the inconsistencies of mal-rules among populations and the unsystematic nature of mal-rules. Besides, the development of mal-rules depends much on the developers' decisions on the levels of abstraction made to the errors. This complicates the situation. However, the most difficult part comes from the breakdown of the old belief that "once a bug has been accurately diagnosed, an instructional prescription follows naturally" (Pintrich, Cross, Kozma, and

---

<sup>30</sup> Errors.

McKeachie, 1986). Sleeman (1987) discovered that "even though [his system] has a model for a student's problem-solving it has not so far proved possible to remedy very effectively." Because of the failure of the mal-rule approach, Self suggested using the student model to represent student's beliefs and that students are provoked into considering the justifications and implications of their own beliefs.

4. "Don't feign omniscience - adopt a 'fallible collaborator'" role". As a precise student model is impossible, Self suggested that machine learning techniques should be employed to infer concepts from examples observed in the students in psychologically way that a student model describing the student's beliefs can be made. The student's model in this case is judgment free and is used for both the ITS and student to refine the student's beliefs. The ITS now becomes a collaborator instead of a tutor because instead of teaching its role is now helping the student to elaborate those beliefs.

Self's suggestions on student models open up a new area for research. When facing with increasingly complex problems, ITS designers now tend to incorporate as many factors as they can into their systems. For example, Merrill, Li, & Jones (1991) in their Second Generation ID Research Program takes into account information about the learner, his or her aptitude, specific goals, motivation, familiarity and other factors, as well as the learner's expressed preferences during the delivery of instruction. In principle, this trend is a way of finding a computer tutor for all students. However, in real situations, it is not easy to incorporate so many factors precisely into a computer system. Further, we can see that it would be difficult even for a human tutor to consider all these factors simultaneously. Of course we can ask more from an ITS, but considering the present hardware constraints and our present understanding

of human factors such as motivation and common sense, it is not realistic to expect so much from an ITS. Besides, by not keeping a precise model does not necessarily mean that the system is not helpful. Self's suggestions above may be useful guides to overcome the various difficulties faced by current intelligent tutoring systems. Exactly how these are implemented will be discussed in Chapter 6.

### The Tutoring Module

This module includes the tutor's knowledge of how to teach effectively, how to present materials, how to discover students' errors and how to correct them. In the words of Woolf (1987), a tutoring module is:

*A system that can teach a student how to solve the problem. It must monitor the student's actions, advise the student about obvious errors, and anticipate future actions based on inferences about the student's current activities. In this interaction the tutor must respond sensitively to the student, must know which activity to suggest, and know how to monitor the student's answers or questions.*

This model comprises three types of knowledge represented in the form of rules used in this module:

1. Didactic knowledge - the tutor's knowledge of how to teach effectively. This includes the sequencing of materials to be taught, when and where in the tutoring process the materials are to be disclosed, etc.



2. Diagnostic knowledge - the knowledge of how to find out the errors of the students. Normally, the errors are revealed by inferring from the students' responses based on these diagnostic rules.
3. Prescriptive Knowledge - the knowledge of how to correct students' errors. These include many possibilities such as re-exposing the didactic materials, pointing out the errors and letting the student redo the exercise or re-answer the question.

A good human tutor would normally use these rules flexibly and intelligently with the students. If an intelligent tutoring system is to communicate effectively with students, it may also have to act in a similar manner. As the present system is not intended for didactic purposes, only the diagnostic and the prescriptive rules are required. Besides the knowledge, another problem is when these rules are to be used. Currently, there are two frequently used methods, the criterion-performance model and the model-tracing method. The following sections will describe these two approaches in more detail and will also discuss how the diagnostic and prescriptive rules could be obtained and how can they be used.

### Diagnostic Rules

Diagnostic Rules represent the knowledge required to diagnose students' errors. In order to do so, the system needs ways to represent the errors in the system so that they can identify them when they occur. However, in order to eradicate bugs (errors) in the student's model (Self, 1988), the system would require a certain understanding of why students make the errors. The following sections describe researches on representations of errors and what causes students to make errors.

Bug Theory

The first systematic study on errors may be that conducted by Brown and Burton (1978) to study students' errors in multi-column subtraction problems. The researchers found that the procedure of doing multi-column subtractions of two numbers can be decomposed as a network of interrelated subprocedures as shown in Figure 3.4 below:

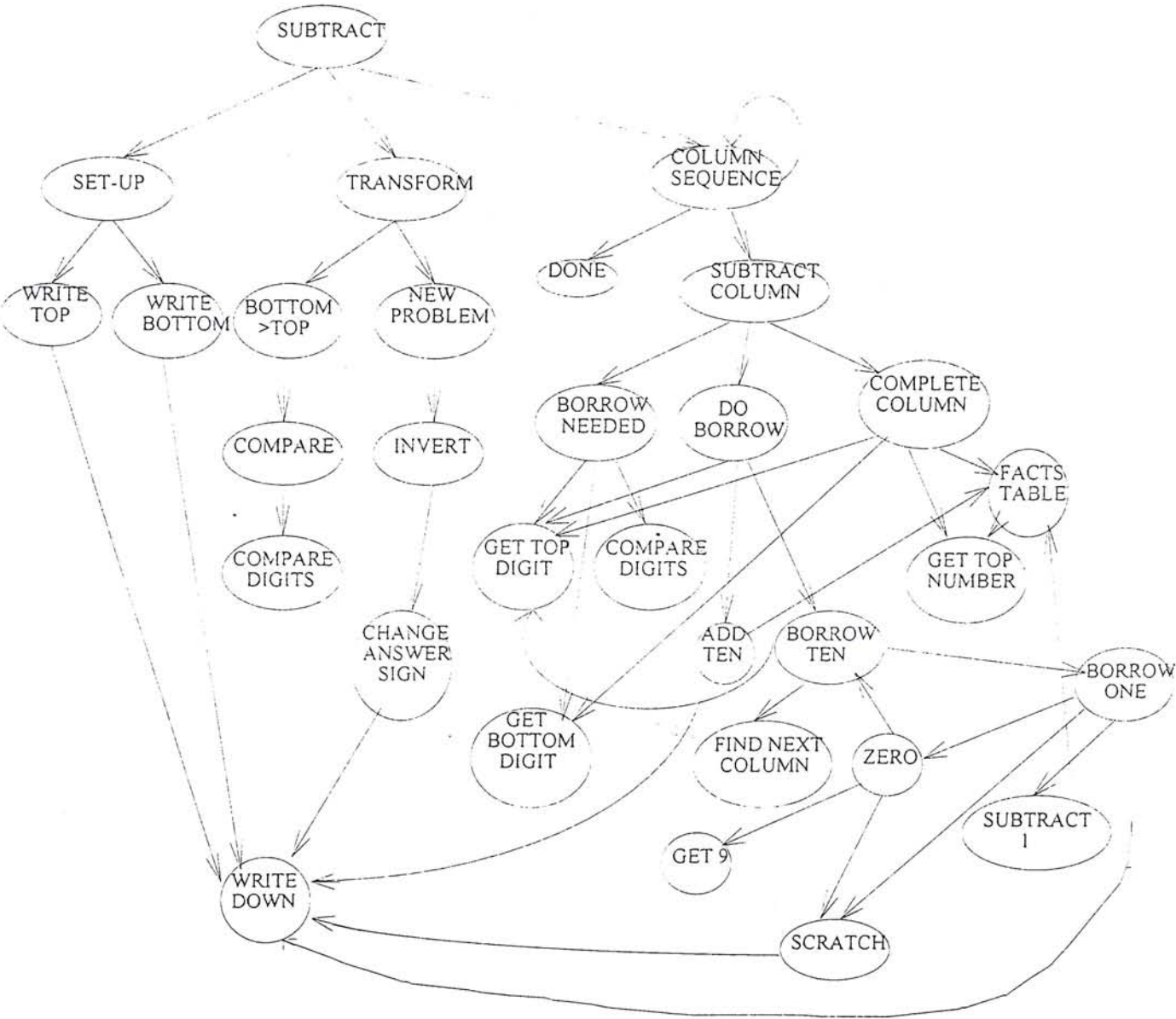


Figure 3.4 A Procedural Network for Subtraction

In Figure 3.4, an arrow indicates the subprocedure that a procedure may use. For example, the topmost node (SUBTRACT) represents the procedure for the subtraction of two n-digit numbers. It may use the procedures for setting up the problem (SETUP), transforming it if the bottom number is greater than the top (TRANSFORM), and sequencing through each column performing the column subtraction (COLUMN SEQUENCE) (Brown & Burton, 1978). The researchers then claimed that any multi-column subtraction can be explained in terms of this network. Based on this network, a computer system which could perform any multi-column subtraction tasks was successfully developed. Besides correctly simulating human behavior in solving these problems, they found that most errors in multi-column subtraction could be simulated in the computer system by slightly modifying one or more subprocedures. An example given by VanLehn (1982a) is as follows:

$$\begin{array}{r} 7 \ 3 \ 2 \\ - 4 \ 3 \ 4 \\ \hline 3 \ 0 \ 2 \end{array}$$

Researchers found the reason to be the students' subtracting the smaller digit in each column from the larger digit regardless of which is on top. Hence there is a buggy version of the procedure (COMPLETE COLUMN) from which the subprocedure (BORROW NEEDED) is missing, so that the student does not recognize the need to borrow. Such a slight modification or perturbation of a correct



subprocedure is called a bug<sup>31</sup> (VanLehn, 1982a), and the theory is thus called the Bug Theory.

### Mal-rule

One of the criticisms of the Bug Theory is that the subprocedures are like black boxes and the exact operations as well as the perturbation are unknown to others. This makes it not suitable for the representation of knowledge that has to be explicitly stated<sup>32</sup>. An alternative approach to the representation of errors is the use of mal-rules. A rule is a statement that describes what action should be done under certain conditions. It is normally in “if-then” form, where the if part describes the condition under which the action, described in the then part, should be taken. A correct rule is a rule that causes the correct action while a mal-rule is one that causes incorrect actions. A set of correct and incorrect rules can be combined to form a production system to simulate the problem-solving processes. Attempts to use production system to simulate subtraction and solving algebraic equations have been successfully demonstrated by Young & O'Shea (1981) and Sleeman (1984). As an example, the Production System<sup>33</sup> developed by Young & O'Shea is described below:

The architecture of the production system developed by Young & O'Shea (1981) has three components:

---

<sup>31</sup> A bug is initially referred to an error in a computer program. Cognitive psychologists borrow this term to represent errors in general. A bug in the bug theory means slightly differently as the perturbation of correct procedure only.

<sup>32</sup> Refer to page 52 for the discussion on explicit and implicit knowledge.

<sup>33</sup> A production system generally means a set of productions but the execution of which requires space for the storage and methods for the selection of productions. Young & O'Shea included all these in the production system.

What Knowledge to Incorporate and How

- A working memory<sup>34</sup> (WM). A set of elements such as ( S EQ M) or (RESULT 5) which are rules to be used .
- Production memory<sup>35</sup>, which holds a collection of production rules in the form of  $C \Rightarrow A$  where A represents the action and C represents the condition that A is to be done. The collection of production rules is shown in Table 3.1.
- A conflict resolution method to determine which rule is be fired (selected) when more than one is applicable.

As an example of how this works, suppose the system is to perform the calculation "74-28=?". With the terms defined as follows, Table 3.2 shows steps and corresponding actions taken by the production system:

	7	4	← minuend	M2	M1
-	2	8	← subtrahend	S2	S1
<hr/>			← difference		

---

<sup>34</sup> Working memory here refers to memory space in a computer for the storage of actions to be done. It is in a way identical to the human working memory that they both deal with things that are currently in the mind (computer), and that they store the knowledge for a short period of time. However, the exact action of each may be completely different.

<sup>35</sup> This corresponds to the human long-term memory. Knowledge is in a sense permanently stored here and will only be placed into the working memory when required.

Table 3.1

Production Rules Used in Young & O'Shea's Production System for Subtraction

Rule Name			
FD:	$M=m, S=s$	$\Rightarrow$	FindDiff, NextColumn
B2A:	$S>M$	$\Rightarrow$	Borrow
BS1:	Borrow	$\Rightarrow$	*AddTenToM
BS2:	Borrow	$\Rightarrow$	*Decrement
CM:	$M=m, S=s$	$\Rightarrow$	*Compare
IN:	ProcessColumn	$\Rightarrow$	*ReadMandS
TS:	FindDiff	$\Rightarrow$	*TakeAbsDiff
NXT:	NextColumn	$\Rightarrow$	*ShiftLeft, ProcessColumn
WA:	Result=x	$\Rightarrow$	*Write=x
DONE:	NoMore	$\Rightarrow$	*HALT
B2C:	$S=M$	$\Rightarrow$	Result 0, NextColumn
AC:	Result 1=x	$\Rightarrow$	*Carry, Result=x

Note. "\*" : actions



Table 3.2

Trace of Production

Step	WM contents (in order of recency)	Rule fired	Action taken, or element asserted
	(PROCESSCOLUMN)		
1.		IN	Do *ReadMandS  Assert (M 4)  Assert (S 8)
	(S8) (M4) (PROCESSCOLUMN)		
2		CM	Do *Compare Assert (S>M)
	(S>M)(S 8)(M 4) (PROCESSCOLUMN)		
3.		B2A	Assert(Borrow)
	(BORROW) (S>>M) (S 8) (M 4) (PROCESSCOLUMN)		
4.		BS1(say)	Do *AddTenToM
5.		BS2(say)	Do *Decrement
6.		FD	Assert  (NEXTCOLUMN)
	(FINDDIFF) (NEXTCOLUMN) (BORROW) (S>M) (S 8)...		
7.		TS	Do *TakeAbsDiff  Assert (RESULT 6)
	(RESULT 6) (FINDDIFF) (NEXTCOLUMN) (BORROW) (S>M)		

The system first realizes that the first action taken should be the process of the rightmost column and hence the rule (PROCESSCOLUMN) is placed into the working memory. The first step is then the execution of the rule (IN) which reads in the values of M (the minuend) and S (the subtrahend). The values read ( $M=4$  and  $S=8$ ) are placed into the working memory which then invokes the system to compare them. The rule B2A is used and the result ( $S>M$ ) is placed in the working memory. This ( $S>M$ ) then invokes the system to add ten to the minuend (AddTenToM), decrement the minuend of the next column (Decrement), and take the absolute difference of S and M. The result 6 for the first column is then given. The process goes on until all the columns are dealt with.

The above example shows how a production can be used to explain the process involved in solving subtraction problems. According to Young & O'Shea (1981), errors can simply be explained by omitting rules from the correct set. In the above example, if the rule (CM) is omitted, then the system will just write down the absolute difference of the digits 8 and 4 without first comparing the digits, an incorrect answer of 4 will then be given for the first column.

### Descriptions Rather than Origins of Errors

The use of mal-rules to represent errors is in some way identical to the use of bugs, except that the condition and action parts of mal-rules are explicitly stated whereas in a subprocedure, only its action is known. Both mal-rules and bugs correctly draw attention to the difference between the "surface manifestation" of a bug and its underlying cause (Young & O'Shea, 1981). However, Young & O'Shea also

criticize this kind of approach as more like "symptoms" than "causes". No attempt has been made to explain how the errors occur. Thus these two kinds of explanation can be just descriptions of errors rather than explanations of why errors occurred. The mal-rule approach leads to two further problems. The first is that since every error is represented by a mal-rule and that many mal-rules occur very infrequently (VanLehn, 1982a; Payne & Squibb, 1990), the number of mal-rules will then be so large that it is not easily handled by most systems. Another problem is that although the term "rule" has a deterministic<sup>36</sup> sense, it is by no means true that the same rule will be applied at every opportunity. According to VanLehn (1982a), student would tinker under the same situation or bug-migrate among different identical situations. The exact meaning of tinkering and bug-migration will be discussed in the next paragraph, but their existence clearly shows that it is possible that more than one rule, either correct or incorrect, coexist under the same condition. Whenever the condition is satisfied, one of the rules will be applied. The applying of the rule may not be easily predicted but clearly the mal-rule approach cannot provide an answer to this phenomenon.

### Origins of Errors

It would be ideal if we could always identify the reasons why errors occur. Once several errors can be identified as being from the same origin, then mal-rules representing these errors can be replaced by the higher-order mal-rule that represents the origin, since they can now be inferred from the higher-order mal-rule. It would

---

<sup>36</sup> A rule has two parts: the condition and the action part. The action is the inevitable consequence of condition part. In other words, a rule is used to represent the meaning that, whenever the condition part is satisfied, the action part will definitely take place.



also be possible that these higher order mal-rules may explain the tinkering and bug-migrating effects found in different situations since some of them may come from the same origins.

### Repair Theory

Researchers have suggested explanations as to why errors occur. VanLehn (1982a), for example, suggested a theory called repair theory stating that when a student gets stuck while executing his possibly incomplete subtraction procedure, he is unlikely to just quit as a computer does when it can't execute the next step in a procedure. Instead, the student will do a small amount of problem-solving, just enough to get "unstuck" and complete the subtraction problem. These local problem-solving strategies are called "repairs" despite the fact that they rarely succeed in rectifying the broken procedure (VanLehn 1982a, 1982b). Repair strategies are task-general methods, most of which are familiar to most subjects. (VanLehn, 1990). According to VanLehn (1990), examples of repair strategies are No-op, Barge-on and Back-up. The strategy No-op means the student simply skips what gets stuck in the impasse. In the Barge-on repair, the student interprets the specification of the procedure in a more relaxed way. Lastly, the Back-up repair means to retreat in order to take an alternative path.

Impasses are caused by incomplete learning. A common source of learning is through examples, exercises and other concrete episodes of problem-solving (VanLehn, 1990) and many students can learn the routines for getting the right answers without the slightest insight into what is going on (Haugeland 1978), A subject reaches an impasse and performs some repairing strategies to overcome the

difficulty. If the repairing is incorrect, then an error occurs. This impasse-repair pair may be stored and becomes a mal-rule which may then be the cause of future errors.

This explanation of errors poses two difficulties: The first is that in VanLehn's view, all errors are caused either by repairing strategies or by previous impasse-repair pairs. In other words, all errors originate from impasses. Although VanLehn claimed that the Cartesian product of impasses and repairs, i.e., the set of all possible impasses-repair pairs may predict all the bugs reported and even those not yet reported (only in subtraction). But we should keep in mind that logically speaking, the fact that the repairing strategies can explain the errors cannot imply that it is the correct explanation. This is just identical to the case where artificial chess players do not necessarily use the human strategies although they can play as well as their human counterparts. We should therefore examine the plausibility of such a theory in more detail before we accept its explanation..

There are several questionable points about the repair theory; the first one being that it is hard to imagine that errors are caused either by procedures in the form of impasse-repair pairs or general problem-solving strategies called repairing strategies. Cauzinille-Marmèche & Mathieu (1988), for example, suggested that students' responses to mathematics problems can be classified into three categories, viz. the arithmetical, the algorithmic and the formal syntactic microworlds. A mathematical microworld is a system composed of objects, relationships among objects, and operations that transform objects and relationships (Thompson, 1987). Further discussion on microworlds will be given in a later section<sup>37</sup>, but if students do

---

<sup>37</sup> Page 120.



respond with different microworlds, it is unlikely that they would always use the general problem solving strategies regardless of which microworld they are in. A more realistic picture would be that between the general problem solving strategies and the domain specific rules used in solving the problem; there may be some rules with domain-specificity lying between the two extremes. These rules, called meta-rules from now on, are rules that are specifically for some purpose and they are grouped together to form the microworlds. When a problem is encountered and there is no suitable domain-specific rule to be used, the students respond with a particular microworld and use those rules to repair the impasse. It is only when no more suitable meta-rule is available that the students would use general strategies to solve the problem.

### Meta-bugs

A similar convention can be found in modeling students' errors in learning English tenses (Giangrandi & Tasso, 1995). In order to model the knowledge representing a generalization of the concept of bug and to describe in a generic form the different kinds of perturbations concerning the domain knowledge, Giangrandi & Tasso (1995) introduced the concept of meta-bug which is the specification of a possible way for altering pieces of correct domain knowledge in order to draw out new possible bugs which are not present in the bug library.

According to Giangrandi & Tasso (1995), the purpose of a meta-bug is to specify how to modify some domain knowledge ( a concept or a rule) in order to model some students' misconception which could not be recognized through the standard bug collection approach. An example of meta-bug is given as follows:



MB1:

During the conjugation of a perfective tense, the auxiliary verb “to have” could be replaced by the auxiliary verb “to be”.

This meta-bug can then be exploited for perturbing different kinds of domain rules and therefore avoiding the definition of many bug rules, one for each tense. An instance for this is given by Giangrandi & Tasso:

Suppose that the student gives the incorrect answer “is gone” instead of “has gone”. In this case, the above example of meta-bug can be exploited for perturbing the following (correct ) rules:

R1:

The present perfect is formed with

- the simple present of the verb “to have” followed by
- the past participle of the verb.

in order to produce the new bug rule:

R1:

The present perfect is formed with

- the simple present of the verb “to be” followed by
- the past participle of the verb.

Whether the above notion on repairing procedures is true depends on future evidence. However, another questionable point is whether impasse is the only source of errors, either an error is formed during repairing an impasse or an error is formed

when a previous impasse-repair pair is used. The argument against this is that although it is true that errors are caused by the means described above, it is also possible that subjects may use rules generalized from similar but not identical situations. Thus the error is caused by the incorrect use of rules but not repairing or using a mal-rule. Evidence for this can be found in previous studies. For example, in studying 25 third-grade students learning multiplication facts, Norem & Knight (1930) found that 91% of the errors were the correct responses to other multiplication problems and about 71% of the errors were the answers to problems that had one multiplicand in common with the given problem. Thus the errors were caused by the incorrect use of correct rules of other problems. Viscuso, Anderson and Spoehr (1989) suggested that interference develops as children learn the multiplication facts, so that most errors are due to associative interference from related problems. If errors are caused by this kind of interference, then it is doubtful whether these subjects would experience impasses when working on their problems.

### Misgeneralization

Misgeneralization has been thought to be a main source of errors (Sleeman, 1989; Matz, 1982, VanLehn, 1990). According to Matz (1982), errors are the results of reasonable, although unsuccessful, attempts to adapt previously acquired knowledge to a new situation. In Matz' theory, whenever an unfamiliar problem is given, the individual's problem solving behavior will include two components. The first is a set of base rules, which is the background knowledge that the student has extracted from a prototype or obtained directly from a textbook. Secondly, he or she will also have a handful of extrapolation techniques that specify ways to bridge the gap between the known rules and the unfamiliar problem. These extrapolation

techniques are the rules they have applied in many situations, and was proven to be very useful. In a sense, this is similar to VanLehn's repairing strategies (1990). It is the use of techniques and the correct prerequisite knowledge that cause the errors. Matz (1982) thought many common errors are shown to arise from one of two processes:

1. inappropriate use of a known rule as is in a new situation;
2. incorrect adaptation of a known rule to solve a problem.

Matz's extrapolation theory differs from VanLehn's repairing theory in two ways; firstly the domain-specific knowledge is included as a source of errors and secondly, the errors are produced by correct knowledge in related areas. The second difference agrees with that suggested by Norem & Knight (1930) and Viscuso, Anderson and Spoehr (1989). For example, students might already know the distributive law which states that when a number A is to be distributed to the sum or difference of two numbers B and C, then it can be expressed as the sum or difference of A times B and A times C. In symbolic form, these can be expressed as :

$$[R34] \quad A \times (B + C) \rightarrow A \times B + A \times C$$

$$[R35] \quad A \times (B - C) \rightarrow A \times B - A \times C \quad (\text{Matz, 1982})$$

where an " $\rightarrow$ " in the above rules means the expression before it can be expressed as that which comes after.

If they are required to simplify expressions in the form of  $\sqrt{A+B}$ , they would probably express it incorrectly as  $\sqrt{A} + \sqrt{B}$  since the multiplication with A is now generalized to include the action of taking square root ( $\sqrt{\quad}$ ). The mal-rule generalized can be expressed as:



[R36] 
$$\sqrt{A + B} = \sqrt{A} + \sqrt{B}$$

Matz (1982) calls this error the generalized distribution and is an example of students incorrectly using knowledge that is correct when used in other situations.

This is not to argue that errors are not caused by impasse-repairing. On the contrary, it is argued that besides repairing impasses, there could be cases where students do not experience any impasses but still commit errors. The errors are caused by incorrect use of rules that may be either correct or incorrect when used in other situations. It is hoped that data collected in this study will be able to be used as supporting evidence. The argument is then impasse-repairing is not the necessary condition for making errors although many errors may be caused by this source.

### Errors caused in the encoding processes

Both the explanations of errors given by impasse-repairing and misgenerating are based on students' internal processes. An impasse is an internal state experienced by the student and misgeneralization begins with an internal representation of a rule (e.g., [R34] and [R35] above) and ends with a new rule ( e.g., [R35]). Both seem to neglect the possibility that errors may be caused when the problem is being encoded. This is also true for production systems that try to simulate the error-generating process. They seem to rely entirely upon internal mental structures and are unable to learn from patterns in the external environment of the system (Payne & Squibb, 1990). Larkin (1989), in her "display-based problem solving", suggested that external display aids the problem solving. Cox and Brna (1995) found that great diversity of external representations was used across subjects. Payne & Squibb (1990) further suggested rule induction from the written input and output of productions perturbed at run time

provides one plausible origin for mal-rules alongside correct versions. External display seems to be an area for errors in addition to the impasse-repairing processes.

### **External Display and Semantic Rationalization**

A further explanation for errors was given by Payne & Squibb (1990) that new (mal-) rules may arise when students attempt to make sense of currently purely syntactic rules. This process is referred to as the semantic rationalization. An example given by Payne & Squibb (1990) is as follows: when simplifying algebraic expressions, students may sometimes make errors like expressing  $3x+2$  as  $(3+2)x$ . This type of error may be represented by the mal-rule [M12]:

$$[M12]^{38} \quad Mx + N \rightarrow (M + N)x$$

According to Payne & Squibb, the error arises because students may semantically rationalize the expression  $Mx+N$  as  $3\frac{1}{2} + 1$  (Matz, 1982) or "three apples plus four". Since  $3\frac{1}{2} + 1 = 4\frac{1}{2}$  and "three apples (i.e.,  $x$ ) plus four (no ' $x$ ') give seven apples", the errors are formulated when students try to interpret the semantic meaning of the expression  $Mx + N$  and rationalize the new expression by using their other life experiences. In this case, errors are not merely generated by syntactically manipulation of the symbols in the expression<sup>39</sup>.

---

<sup>38</sup> This is the original coding used by Payne & Squibb. It is not coded according to the convention used in this study.

<sup>39</sup> For example, rule [R36] is caused by treating the action of " $\sqrt{\quad}$ " as identical to " $\times$ ". There is only symbol manipulation but no interpretation of the meaning of the expressions.



### **Off-line and On-line Errors**

So far we have described four explanations of why errors occur. Although all of them suggest that errors are caused by different reasons, there is no disagreement on how the newly generated error may later affect students' problem solving behavior. According to these studies, errors are first generated either by repairing an impasse, misgeneralizing, misinterpreting the external display or semantic rationalization. These generated errors are then stored as mal-rules which may later affect students' behavior. Errors happen whenever these rules are used.

Hence if we could categorize errors according to the time they are generated, we can say that there are two types of errors: the off-line errors and the on-line errors. Off-line errors are those caused by mal-rules generated some time before, while the on-line errors are generated on the fly. An off-line error and an on-line error may come from the same origin, but the chances of their being used are different. Newly generated rules may have less chance of being used while those used frequently will have a higher chance (Payne & Squibb, 1990). However, their chances should also depend on whether there are some other rules competing to be used. In the next section, we will discuss competing of rules.

### **Competing of Rules**

Quite often two rules, either correct or incorrect, may coexist under the same situation. Tinkering, which is the case that students sometimes making one error and sometimes another within the same situation (VanLehn, 1982a), is a clear example of the coexistence of rules. Also, even adept problem solvers in arithmetic as ourselves may sometimes makes error like writing  $2+3$  as 4 although we are quite clear that the



answer should be 5. This is another example since we may have one rule saying that  $2+3$  is 5 and the another saying that  $2 + \text{something}$  is 4, but due to cognitive load or any other reasons, you forget what the “something” is.

How are we going to explain why we choose one but not the other? Payne & Squibb (1990) simply described this in terms of probability. According to Payne & Squibb, the choosing of which rule depends on how strong these rules are linked to the situation they are needed. Stronger rules have a higher probability and weaker rules a lower probability. The strength of a rule is determined by its frequency of use and its semantic rationalization with other knowledge.

### **Perception of problems and errors**

It is plausible to use probabilities to predict which rule is chosen, but it is certainly not possible to explain why this particular rule is chosen at that moment by using probabilities only. To get more insight into this process, it is worthwhile to look more closely at the tinkering effect, which says that one rule is chosen at one instance but another rule at another instance.

The earlier discussions on external display (Larkin, 1989) and semantic rationalization (Payne & Squibb, 1990) suggest that errors may be caused during the encoding process. However, both of these two still use the impasse-repairing approach to explain the errors with the exception that the repairing is not done by using general heuristics. Instead, justification or semantic rationalization based on the external display are used. They both neglect a possibility that students would sometimes perceive a problem very quickly and arrive at a decision on what to do next without deliberating the reasons behind the decision.

A problem has to be perceived or encoded before it can be solved (Polya, 1957; Scandura, 1974; Newell & Simon, 1972; Lee, 1981). If deliberative encoding such as semantic rationalizing can be a cause of errors, there can be no reason why some quick and automatic perception which is acquired through practice would not cause errors. It is this kind of fast and automatic misperception of a problem that cause the student to make an error by using rules that are thought to be appropriate in the perceived situation.

The example given by Payne & Squibb (1990) to show the effect of semantic rationalization may well be an example of errors caused by perception. How this example can be used to illustrate the perception error will be discussed in the next paragraph. However, in order to help students to correct their errors, the error caused by communication failure between the student and the problem should be identified and minimized. If we overlook the process of communication failure and repair in an ITS, we may find ourselves trying to solve the impossible problem of anticipating and programming all possible situations of knowledge misunderstanding (Douglas, 1991).

The example given by Payne & Squibb (1990) on semantic rationalization is set out below. Two mal-rules named separately as [M12] and [M13] were identified as follows:

$$[M12]: Mx + N \rightarrow (M + N)x$$

$$[M13]: Mx + N \rightarrow (M + N)$$

The two rules occur in exactly the same circumstances, and both are relatively frequent. But they found that [M12] appears 102 times whereas [M13] appears only 49

times. According to the researchers, [M12] is derived by over-generalizing from

$$Mx + Nx \rightarrow (M + N)x$$

to  $A + B \rightarrow (\text{number} - \text{part} - \text{of} - A + \text{number} - \text{part} - \text{of} - B)x$

and [M13] from over-generalizing from

$$M + N \rightarrow (M + N)$$

to  $A + B \rightarrow (\text{number} - \text{part} - \text{of} - A + \text{number} - \text{part} - \text{of} - B)$

Both can be over-generalized syntactically<sup>40</sup> from other correct rules. However, as [M12] is more sensible that it is analogical to say " $3\frac{1}{2} + 1 = 4\frac{1}{2}$ " (Matz, 1982) or from natural language constructions, like "three apples plus four give seven apples". In other words, [M12] may be generalized semantically. Hence [M12] would appear more frequently than [M13].

Payne and Squibb (1990) did not specify when this semantic rationalization happens. It is possible that the rationalization occurred some time before and is now a mal-rule that causes the present error. Even if this is true, there must be a first time for this mal-rule to be formulated. Assuming now, a student is asked to deal with an expression in the form of  $3x+4$ , which he or she has never met, what will this student do? Will this student sit down, think for a while and say, "I don't know how to do it. But it looks similar to the expression 'three apples plus four', and since three apples plus four gives seven apples, the answer should be  $7x$ "? In this case, the student can be said to be doing some semantic rationalization since he or she has no idea

---

<sup>40</sup> i.e., by simply manipulation of symbols.



whether this is right or wrong. The use of the apple example is just to rationalize the answer given.

On the other hand, the student might perceive the situation as " $M$  of something plus  $N$  (of that thing)" and then the result comes out when the following rule, which he or she has used many times successfully, is used automatically:

[R37] If  $M$  of something plus  $N$  (of that thing) is to be simplified, then write it as  $(M+N)$  of that thing.

In this case, the student has not experienced any doubt and the error is caused by a rule that should be used in other situations. Thus, although errors may be generated from the same semantic reasoning and the same situation, different mental processes would cause different types of errors. It is only through the investigation of the error-generating process that evidence of this perceiving error can be revealed. Although this kind of perceiving error may well have happened in previous studies, it has never been reported. The reasons might be because in previous studies, there are limited ways of looking at the problem in restricted areas such as multi-column subtraction or solving linear algebraic equations in one variable, hence the effect of perception in producing errors was overlooked.

### **Section Summary**

Basically, previous studies agreed that errors are caused by repairing impasses. It is only the repairing strategies assumed that make the difference. According to these studies, repairing can be done either by using general heuristics, misgeneralizing, guessing based on external or semantic rationalizing. Although impasse-repairing pairs cause errors, only new errors are being generated by such pairs since impasses

happen when there is no suitable rule available. After a new error is generated, it would be stored in the form of a mal-rule with the problem situation that generate it being the condition part. Next time when the same situation happens, no more impasse has to be repaired since there is a rule available. Since the rule used is an incorrect rule, an error thus occurs. This type of error is referred to as off-line errors since the mal-rules representing the errors are generated beforehand. Correspondingly, the errors generated immediately after the repairing are called the on-line errors.

Although previous studies can clearly differentiate between the on-line and off-line errors, none of them can recognize the possibility of two different reasons for the off-line errors. In all these studies, students are assumed to be able to perceive the problem situations correctly and that off-line errors are caused by the use of mal-rules. It is argued in this section that students may sometimes incorrectly perceive the problem situation and then use rules that are correct, but only in the perceived situation, to solve the problem. Errors are thus generated but these errors are not related to mal-rules and students making such errors do not experience any impasse. Evidence of such errors will be reported in Chapter 6 of the present study.

### **Prescriptive Knowledge**

Prescriptive knowledge is the knowledge on how to help the students based on the diagnosed errors. Basically, the work done may be grouped as answers to the following questions:

1. When should the students be corrected? Should a student should be allowed to freely explore the problem during the solving process or should errors be immediately corrected to prevent it from becoming a stable error?



2. How should student be corrected? Should they be given more practice and hints, or should the content containing errors be retaught?

### **When to Remedy**

The first question is related to the earlier discussion on the criterion-performance model method and the model-tracing method<sup>41</sup>. A computer system is said to be employing the criterion-performance model method if all the possible paths to solve a problem are pre-generated and stored up as models, with no regard as to whether these paths will lead to correct answers or whether each step is correct. When the problem is given to a student, he or she is allowed to complete the problem-solving without any interruption from the computer tutor. The completed process is then checked against the set of models to see if there is a match. If a matched model that leads to the correct answer can be found, then the student is considered to have done the problem correctly. On the other hand, if the matched model leads to an incorrect answer, then the mal-rules used to generate this model can be used as the basis of remediation to the student. If there is no matched model, the system is considered as unable to diagnose the student's problem solving process. One such example is PIXIE developed by Sleeman (1987) described earlier<sup>42</sup>. On the other hand, systems employing the model-tracing method does not pre-generate the models. Instead, at each state (step) of the process, the system infers the learner's internal state by matching his output with the problem state generated by using ideal and buggy rules (incorrect rules). Instructions will be given according to this inference.

---

<sup>41</sup> Page 75.

<sup>42</sup> Page 75.



### **Criterion-performance Model**

Thus the main difference between the two methods is the amount of freedom given to students to explore during the interaction process. In the former method, instructions will only be given at the end of each problem, thus students are allowed to flounder freely. It is possible that students would make errors during floundering, but it is also possible that a student would discover his or her errors and recovers them. Hence floundering may be good experience for the student. According to Sleeman and Brown (1982),

*"Some floundering can be vitally important. The crucial meta-skill of knowing when one's floundering is useless can only be discovered by trial and error. The subtlety of separating potentially productive exploration from useless wanderings only points to the challenges of constructing sensitive and well-motivated tutorial principles."*

In PIXIE (Sleeman, 1987; Moore & Sleeman, 1988), students are allowed to freely explore when solving problems. PIXIE has been implemented to involve three separate phases: the off-line, or model generation, phase; the on-line, or tutoring, phase; and the analysis phase. Models that incorporate typical mal-rules (errors) in addition to correct knowledge are generated in the off-line phase. During the tutoring phase, a student is given a range of tasks to solve and the answer is compared with the models generated in the off-line phase. Remediation based on the interpretation of the student's individual problem-solving strategies and consistent bugs (errors) will then be given. During the post-interaction analysis phase, undiagnosed errors are examined and, if consistent, added to the existing domain knowledge base (Moore & Sleeman,

1988). This model of remediation is thus called the model-based remediation (MBR) since the remediation is based on the student models generated.

### **Model-tracing**

In contrast to keeping a large number of models, a system that employs the model-tracing method does not keep even one complete model of the problem solving process. For example, in *The Teacher's Apprentice* (Reiser, Anderson, & Farrell, 1985), there is no criterion-reference model. Instead of using models, a student's answers will be checked at each step against a set of correct rules and mal-rules. Remediation will be immediately given whenever the student is found to be using a mal-rule. Such model-tracing technique has the advantage of not having to keep a large number of models and thus there is no need for off-line model generation. Besides, as the checking is done at each step, it can be done much faster than that in the criterion performance model, in which many models each with a large number of different paths are to be compared. In this way, this model-tracing method is better than its counterpart.

### **Repairing and Kept in Correct Path**

Although a model-tracing system is, in many ways faster than those using criterion performance approach, this does not mean that it is entirely satisfactory. A major criticism of this kind of systems is that it always keeps the students on the correct path leading to the solution, thus ignoring the principle that making errors or floundering may be an important part of learning. As stated by Sleeman and Brown (1982), floundering would help students to develop meta-skills of knowing. Depriving

students of the chance to do this may detract educational value from this kind of system.

Whether meta-knowledge will be developed is not known yet, but there is little evidence supporting that floundering is important to students. On the contrary, the model-tracing approach is in line with our discussion of errors. Errors are generated by repairing impasses. According to VanLehn (1982a), impasses repairing is a kind of problem solving process similar to the problem solving process of students. If impasses-repairing would become a mal-rule that causes later errors, so would the errors made by the students when they are floundering in the problem solving process. Keeping students on the correct path is one way to avoid the occurrence of such mal-rules. Hence this model-tracing approach should be the more effective way to teach students to correctly solve a problem although there will be a possibility that students are deprived of the chances of developing meta-knowledge.

After considering the two different approaches, a better choice of the tutoring strategy would be the Model-tracing Method, which in the words of Fox (1991), is one in which student errors, and in particular tutor correction of student errors, are kept to a minimum. This would be identical to the tutoring process of a human tutor; whenever the learner deviates from the ideal (norm) path, a sequence of increasingly specific hints would be given (Kamsteeg & Bierman, 1991). The whole idea is that...



*the tutor always provides a safety net around the student, so that, if she shows signs of not being able to answer the question, the tutor offers a resource for answering. If the student shows signs of not seeing the import of a question for the problem at hand, then the tutor steers the student towards seeing the connection. (Fox, 1991)*

Similar strategies were employed by many researchers in the systems they designed (Lewis, Milson, & Anderson 1987; Kamsteeg & Bierman, 1991; Wang & Garigliano, 1992; Matsuda & Okamoto, 1992).

### **How to remedy errors**

Another problem in prescriptive knowledge is how to remedy students' errors. Errors are represented by mal-rules. According to Payne & Squibb (1990), several rules, either correct or incorrect (mal-rule) coexist under the same condition and that one of the rules will be selected. The probability for the firing (selecting) of each rule depends on its strength. Stronger rules will be fired more frequently than weaker rules. Strengths of rules are determined by their frequencies of use in the past and their semantic rationalization. Thus rules used frequently in the past and rules that can have some account of why the manipulations are valid, in terms of the meaning of the symbols on which they operate will be more likely to be used in the present.

If this is the case, then an obvious way to help students correct their errors is either to decrease the strengths of the mal-rules or to increase the strengths of the correct rules with respect to the given situations. Methods to do this is by either practising more the correct rules or linking the correct rules to students' previous knowledge. Example of latter can be found in Appendix E.

In order to have a clearer picture of the relationship between the probabilities of errors and the strengths of rules, Lee (1994) built a neural network model to simulate a mathematics problem solving processes. Using this model, the different stages of a problem solver, starting from a novice to a consistent user of mal-rules, and finally to an expert, can be simulated. These different stages were suggested by Sleeman (1985) as the pattern of maturation during their understanding of a topic:

UNPREDICTABLE -> CONSISTENT USE of MAL-RULES -> CORRECT

The pattern of maturation can be simulated by first increasing the strength of links joining the input and the incorrect rules, then increasing the strength of links joining the input and the correct rules. As the process of changing from a consistent mal-rule user to a correct rule user is exactly the process of correcting an error, and this process can be done by increasing the strength of the correct rule, Lee (1994) thus argued that errors can be removed by practising the correct rules. The following sections summarize other research findings on prescriptive strategies.

### **Reteaching and Model-based Remediation**

One of the major differences between the designing principles of the traditional CAI and ITS is their different approach to remediation. In traditional CAI, the remediation is done by merely reteaching the materials in which the subject's errors are found. For example, when a student makes an error in solving algebraic equations, the rules for solving equations are retaught and the subject has a chance to repeat the problem. However, in ITS a basic assumption is that errors are systematic and that the diagnosis is more difficult than the remediation. Once an accurate model of the student's error has been inferred, it is then relatively straight forward to use that



model to direct a remedial dialogue (Sleeman, Kelly, Martinak, Ward & Moore, 1989). Sleeman et al. referred to this method of providing procedurally orientated remediation of specific errors found in a student's solution before reteaching a correct strategy as the model-based remediation (MBR). From the view of ITS designers (Brown & Burton, 1978; Resnick, 1984; Macnab & Cummine, 1986), this MBR works better than the traditional reteaching of CAI.

Empirical research, however, shows contradictory findings. Swan (1983) reports that a conflict approach (by pointing out errors made by students and demonstrating their consequences) is more effective than simple reteaching. On the other hand, no difference between the error-specific remediation and reteaching has been reported (Bunderson & Olsen, 1983; Martinak, Schneider, & Sleeman, 1987; Sleeman, Kelly, Martinak, Ward, & Moore, 1989). The latter results have had a great impact on the design of ITSs. As reteaching was mostly done by reteaching in the "classical" computer-assisted instruction (CAI), while ITS relies mostly on the error-specific approach, the result would imply that CAI is as effective as ITS although the latter is more cognitive orientated.

One may attribute the similar effects resulting from the two remediation methods to the lack of attention on the part of the student. If the student does not pay attention during the remediation processes, the choice of method would be immaterial since no learning would actually occur. However, empirical findings by Sleeman, Kelly, Martinak, Ward, & Moore (1989) show that subjects do better in the posttest than in the pretest. Hence, it is clear that learning does occur and that lack of attention cannot be the reason for the similar effects resulting from the two methods.



### **Cognitive Engagement and Cognitive Dissonance**

The error-specific approach was referred to as the model-based remediation (MBR) since the remediation is based on a student model inferred from what the student used in solving the given problem. As the MBR method focuses on the student's specific errors, it should be more effective than Reteaching Only since the student's cognitive load would be much reduced by focusing only on the part where the error occurred. This might be what Sleeman et al. (1989) assumed initially, leading them to believe that similar results obtained from MBR and Reteaching Only came about because the students were not cognitively involved in the remediation processes. They then tried to add two additional components, namely inducing cognitive engagement and inducing cognitive dissonance, to the model-based remediation and then comparing the effects of these two new methods with that of Reteaching Only. Cognitive Dissonance was created to students by demonstrating the unsound nature of the student's incorrect method (Macnab & Cummine, 1986) while Cognitive Engagement was done by having students verbally repeat the correct procedure back to the tutor (Sleeman, Kelly, Martinak, Ward, & Moore 1989). However, results of both the posttest and retention test again showed no significant difference between the two conditions. Finally, arising from the suspicion that the existence of unstable errors might dilute effects caused by the two remediation methods, the researchers focused only on stable errors and attempted to find which of the methods would be more effective in reducing errors. Again, no significant difference was found.

Two reasons were given by Sleeman et al. (1989) to explain the little difference in effects resulting from the two remediation methods: viz. that MBR and

Reteaching Only are very similar and that students in reteaching generated their own MBR. Further investigation is needed to verify these explanations are correct. However, a method better than the traditional Reteach Only is required. Otherwise, there will be no theoretical basis for the tutoring methods currently employed in intelligent tutoring systems.

While both the methods MBR and Reteaching Only focus on procedurally correcting students' errors<sup>43</sup>, mathematics educators have already pointed out that meaningful learning based on the understanding of concepts related to the procedures is more beneficial to students than learning the procedures by rote (Davis, 1984). Even with the two additional components, Cognitive Engagement and Cognitive Dissonance, the tutoring processes are still procedural orientated: students are only required to rehearse the rules in the Cognitive Engagement condition. It is doubtful whether this would induce students to actually engage cognitively in learning the rules, not to mention using these rules in later problem solving tasks.

On the other hand, in the Cognitive Dissonance condition, dissonance is induced by getting the students to notice the incorrect results generated when their solutions are substituted into the original equations. The idea may be that students would be more conscious of the errors made when dissonance is induced. However, though the students may be more aware of their errors, if there are no suitable means to help them to use the correct rules, this induced dissonance could do no more than point out their errors. It is commonly acknowledged that telling students they are wrong does not help them to correct their errors.



### Connection to Conceptual Knowledge

Many researchers (for example, VanLehn, 1990) agree that students generally rote-learn mathematics algorithms without connecting them to the underlying semantic information. Systematic errors would then occur when the algorithms are misused (Resnick, 1982). Sternberg (1985) also pointed out that reasoning may consist of the manipulation of mental models<sup>44</sup> that correspond to internal analogues of scenes of actors and errors made because people fail to consider all the possible models of the premises. Both semantic information and mental models are, in a broad sense, conceptual knowledge related to the procedures where errors occur. Both researchers seem to suggest that increasing conceptual knowledge would be an effective way of reducing systematic errors.

Concerning mathematics, in particular algebra, Cauzinille-Marmeche and Mathieu (1988) suggest that students encounter difficulties when assimilating the rules for rewriting expressions because these rules seem to them an arbitrary collection, independent of each other and not connected to their previous knowledge. These difficulties may be derived in part from a tendency in students and teachers to treat algebra as a purely formal system, without reference to the number relationship and situational constraints that give it referential meaning. Evidence from research by Hinsley, Hayes & Simon (1976) shows that students can have considerable difficulty in relating algebra equations to basic ideas such as equivalence and functional relationships.

---

<sup>43</sup>Although in MBR, an accurate model of a student's error has been inferred, the remediation provided is still procedurally orientated.



### **Microworlds**

In studying students' comprehension of literal or numerical expressions of the form  $a+b+c$ , with or without brackets around either the  $ab$  or the  $bc$  portion of the expression, Cauzinille-Marmèche and Mathieu (1988) argued that students' responses could be classified into three categories viz. the arithmetical, the algorithmic and the formal syntactic microworlds. In the first one, the expressions to be examined are analyzed as a chain of transformations applied to an initial quantity. In the second one, the expressions to be compared are examined by actually calculating the quantity, and if the expressions are literal, letters are replaced by numbers. Finally, in the formal syntactic microworld, the expressions are analyzed as chains of symbols. The researchers found that students, especially young ones, spontaneously refer to distinct microworlds which are not necessarily related to each other and do not form a coherent structure. They then suggest that inducing students to change their representation and to establishing links between different representations can help them understand and use the newly introduced algebraic rules. This instructional strategy was supported by experiments done on electricity and electronics (Gentner & Gentner, 1982) and elementary mathematics (Resnick, 1982).

### **Practice and Conceptual Links**

While there is no conclusive agreement on whether MBR or Reteaching Only would be more beneficial to students, it seems that the work done by Cauzinille-Marmèche & Mathieu (1988) suggests a new direction for the remediation process.

---

<sup>44</sup> A mental model is the cognitive representation of some particular content domain as defined by Payne (1988). The definition might be a little bit loose but serves the present purpose.

This strategy was also supported by Lee (1994). In building a neural network to explain the process involved in simplifying mathematics expressions, Lee suggested that during the process, several rules may be available to the student, from which he chooses only one. The choice depends on the relative strengths of the rules; stronger ones are more likely to be chosen. Hence, to correct errors, or more explicitly, to correct the procedure that produces errors, this procedure must be unlearned. According to Lee (1994), knowledge can only be unlearned through the learning of a new procedure with prerequisite conditions identical to the one to be unlearned. By increasing the strength of this new procedure, it has more chance to be chosen, at the same time reducing the chance for the old procedure to be selected. In this sense, the old procedure is unlearned. The learning of a new procedure should be done in two ways: practising and building more links to the subjects' previous knowledge. Building more links enables the correct procedures to be more easily accessed, while practice increases the strength of these new procedures.

Here is a simple example. When students are asked to evaluate the expression "log 5" by using the logarithms of special values such as  $\log 1 = 0$ ,  $\log 2 = 0.3310$ ,  $\log 3 = 0.4771$ ,  $\log 10 = 1$ , etc., two possible responses are "log (2+3)" and "log(10/2)". The choice will depend on the students' past experiences and the cues they pick up from the given problem. If a student selects the wrong response "log (2+3)", the way to correct this error is to let the student practise the correct one (i.e., "log(10/2)") and at the same, try to link this with the student's past experience (For example, if the expression is expressed as "log (10/2)", then it is possible that the problem can be solved by using " $\log 10 = 1$ ").



While both MBR and Reteaching allow subjects to have more chances to practise, there is no direction given for what content to practise. It is possible that the mal-rules as well as correct rules will be rehearsed which then reduces the effects of remediation. However, the most important consideration is that neither method can help students to build up more links to the previous knowledge. It is therefore understandable that neither method can produce better tutoring effects. The situation cannot be improved even if cognitive engagement and cognitive dissonance were added to MBR.

This argument leads to an assumption that linking students' errors to their previous knowledge may be a good way to helping students correct their errors. As errors are pieces of incorrect knowledge, the link of which to previous correct knowledge would normally induce dissonance. Chapter 5 will describe how a new tutoring method, called Conceptual Dissonance<sup>45</sup> (CD), is designed based on this assumption. The effect of this method will then be compared with that of MBR and Reteaching Only. Additionally, in order to substantiate the assumption pointed out by Lee (1994) that practising enhances learning of rules, the effect of practising was separately measured. It is hoped that a clear direction on strategies to be used in the remediation process of the present tutoring system can be identified.

### **The Communication Module**

This module deals with the interaction of the human and the computer tutor. Normally, its work includes translating the human language into computer language



and translating computer language into that which human can understand. Together they form the input and output components of the system respectively. The translating of human language is not an easy task since human language is not well defined and is sometimes even illogical. To tackle this problem, some intelligent systems such as Meno Tutor (Woolf, 1987) use a kind of restricted language in which the vocabulary consists only of a limited number of terms and the grammar used is strictly defined. With this kind of language, the computer could then understand what the human user enters and reacts suitably.

Another types of systems solved this problem by displaying some icons on the screen so that the users can choose their actions by simply clicking the appropriate icons with the mouse. An example is The Teacher's Apprentice (Reiser, Anderson, & Farrell, 1985). As only a limited number of icons can be displayed on the screen, the latter method is only capable of handling simple interactions. However, even the former limited-language method cannot allow complex human-computer interactions. It only works well in domains such as mathematics or computer programming.

The design of the interface is important. As pointed out by Reiser, Anderson & Farell (1985), "the design of the interface can make or break the effectiveness of a tutor, regardless of how clever the design that went into the guts of the underlying system". The same researchers also pointed out five critical features of an effective interface for tutoring based on their experiences in interface designing. According to them, an effective interface should:

---

<sup>45</sup> Different from cognitive dissonance which is done by getting the students to notice the incorrect results generated when their solutions are substituted into the original equations

1. be as easy to use as possible;
2. have a structure or representation that is as congruent as possible to the underlying structure of the problems to be solved;
3. be highly interactive and provide as much information about intermediate problem-solving states of the learner as possible;
4. have the ability to notice low-level errors as they occur; that is, it should continuously monitor the student's input; and
5. have the ability to vary working memory load. This can be accomplished by giving the student ready access to problem-relevant information and minimizing the number of parts of the screen that have to be attended to and integrated.

Electronic Homework is intended to be used by school students who may not be good at typing. For these students, it would be easier for them to use the mouse as the input device. If all the symbols required in a logarithmic expression can be displayed on the screen so that students can simply use the mouse to drag and paste the required ones in their own expression, the troublesome task of typing in the expressions may be much reduced. Besides, this would also reduce their chances of making low-level errors such as missing brackets.

To reduce students' working memory load, the required formula as well as other given values can be displayed on the screen for their easy access. Further, the screen can be divided into three parts: one for displaying the student's steps in solving the problem, the other for displaying feedback to the student and the third one for displaying the formula and constants. In this way, the student would immediately



know where to concentrate during the problem solving process. Finally, as in some older software, mathematics expressions are displayed in the form that students may not be accustomed to. For example, the product of 2 and 3 would be expressed as  $2*3$  and the 3<sup>rd</sup> power of 5 would be expressed as  $5^3$ . This might be needed if the keyboard is used as the input device. Now if a mouse clicking on icons displayed on the screen is used as the input method, this kind of convention should be modified to help students' understanding. Arising from these suggestions, the exact design of the interface will be given in Chapter 5.

## Chapter Summary

This chapter discusses what knowledge should be incorporated into Electronic Homework and how this could be done. Four modules, the expertise, student, tutoring and communication modules, are used to incorporate the domain knowledge, student model, tutoring knowledge and communication knowledge respectively. The knowledge pieces, whether correct or incorrect, are stored in the form of rules or mal-rules respectively. The different types of knowledge involved and their relationships can be found in Figure. 3.1 at the start of this chapter.

To express explicitly the knowledge involved in Electronic Homework, every knowledge piece is expressed in the form of rules. Some of these rules called correct rules are used to represent correct knowledge pieces and can be found in the expert module. Some other rules are used to represent incorrect knowledge pieces, and are thus called mal-rules (incorrect rules). These rules can be found in the student module. As mal-rules are incorporated in Electronic Homework, the system is thus said to be



using the bug-identification student model instead of the overlay model in which only a subset of the expert's knowledge is included.

A large portion of the chapter has been devoted to the discussion of whether criterion-performance model or model-tracing method should be employed as the tutoring strategy. Two reasons are raised to support that the view the model-tracing method should be employed. First, the current system is designed to be used by personal computers. Such computers may not have sufficient memory to maintain a catalogue of all possible models and at the same time generate a quick enough response to the students that would be required by the criterion-performance model method. Second, it is believed that although the model-tracing method deprives the students of chances to develop problem solving meta-knowledge, the method does reduce their chances of making errors and hence avoid the formation of systematic errors. The model-tracing method is thus selected.

Another major issue discussed in this chapter is errors. While bugs and mal-rules are thought to be only representations of errors, the different possible reasons for errors to be generated, including the impasse-repair pairs, misgeneralization, external-display and semantic rationalization are discussed. Although all such reasons are found reasonably explained some of the errors, it is argued that some errors are better explained as the students have misperceived the problem and thus use the rules that are correct in the perceived problem situation to solve it. Evidence supporting this is to be reported in Chapter 6.

On how the students are to be taught during the problem solving process, two methods that are used in other studies, Reteaching Only and Model-based

Remediation, are compared with a method called Conceptual Dissonance, The last method attempts to induce dissonance between students' errors and is argued to be the best of the three.

Finally, the discussion of the communication module leads to the conclusion that an icon-based interface would be better for the students to communicate with the computer tutor. All the suggestions made in this chapter will be tested by using procedures shown in Chapter 5 while the results will be shown in Chapters 6 and 8.

## ***CHAPTER 4***

# **PROBLEM COMPLEXITY AND INDIVIDUAL DIFFERENCES**

### **Overview**

Following the discussion on knowledge represented in Electronic Homework, this chapter discusses how the difficulty level of a problem and how individual differences among students can be measured. In chapter 1, it was proposed that problems given to students should be arranged in increasing degree of difficulty to maintain students' motivation and the measure of problem difficulty should be obtained<sup>46</sup> before presenting the problems to the students. In this chapter, various measures of problem difficulty are discussed which are all related to a new measure called problem complexity. This problem complexity is a measure of the complexity of a problem in terms of the number of steps required to finish the problem, numerical complexity and other factors. Also, this measure can be obtained before the problem is presented to students and is thus a better measure of problem difficulty if it can be

---

<sup>46</sup> Page 27.



proved that it can actually measure the problem difficulty to a reasonable degree of accuracy.

It is proposed that the effectiveness of Electronic Homework will be tested among students and it is conceivable that some students may benefit more than others from using this system. We need, therefore, to find out what characteristics of the students are related to the varying degrees of usefulness of the system. In this way, the system can be made more adaptable to those who benefit less. At the same time we can acquire a better understanding of the usefulness of intelligent tutoring systems like the present one.

---

Electronic Homework is designed to help students to do their homework. It would be logical to assume that students would be better motivated in their work if the problems in the homework assignment were presented in increasing degree of difficulty. Hence, we must first determine how to measure the difficulty level<sup>47</sup> of a problem before the system can be really used with students.

Traditionally the difficulty level of a problem is measured by a ratio called the item difficulty ratio which is the ratio of the number of respondents answer correctly to the total number of responses to the problem (Gronlund, 1981). This gives a convenient measure of problem difficulty since it is not difficult to find out the two quantities required for the calculation once the problem has been administered to students. However, a convenient measure does not necessarily mean that it is an accurate measure. As a matter of fact, it is argued here that two criteria should be used

to decide on the appropriate measure of problem difficulty to be used in the present study. The criteria are that the measure should indicate how much cognitive effort is required from the students and that it should be obtainable before any kind of test using the problem can be administered. The following two sections describe reasons supporting these two criteria:

## Cognitive Difficulty Or Simple Item Difficulty Ratio

The desired measure should be a measure of how much cognitive effort is required rather than simply deciding whether the problem is difficult or not. According to Mason, Zollman, Bramble and O'Brien (1992),

*This definition (difficulty in terms of item difficulty ratio) implies an easy item also would be easy in terms of the cognitive challenge it presents to a respondent. Such a conclusion might be incorrect. Easy items might be answered correctly for the wrong reasons (e.g., there may be a wording clue that points to the correct answer, or the answer might be given on the basis of automatic grammatical responding rather than thoughtful reply); similarly, a difficult item might not represent a difficult concept, but it might be so poorly phrased as they encourage incorrect responding.*

Thus, there may be many reasons why a problem is difficult. Poorly phrased or misleading problems are difficult but do not require the student to understand difficult concepts or performing complex calculations. Once these problems are rephrased or

---

<sup>47</sup> How difficult a problem is.

clarified, the problem may be easily solved. For example, Linville (1970) reported that students might solve more problems when the syntax of the problem statement is not complex. These problems do not offer any cognitive challenge to the students and it is thus. meaningless if problems are arranged in terms of such item difficulty ratio. This assumption was supported by the evidence given by Newman, Kundert, Lane, and Bull (1988) that students' scores are not improved in a test of multiple choice items when these items are arranged in increasing order of statistical difficulty (item difficulty ratio).

Another approach to order problems suggested by Newman, Kundert, Lane, and Bull (1988) is by using cognitive difficulty. The researchers categorized 40 multiple choice examination items in educational psychology into three levels of the Bloom's taxonomy: knowledge (15 items), comprehension (11 items) and application (14 items). Items belonging to different levels were then considered as of different levels of cognitive difficulty. Results showed that students would obtain higher scores in harder problems when the problems were arranged in increasing cognitive order (knowledge, comprehension, application). No such effect was found in medium and easy problems. Cognitive difficulty is thus a possible measure for the basis of ordering problems in tests.

## **Difficulty Level Obtained Before Test**

### **Administration**

Item difficulty ratio has to be obtained after a test of such items is administered. If the questions used in Electronic Homework are those already done by



students, item difficulty ratio may still be a possible means of measure provided it can really represent the cognitive difficulty. However, as Electronic Homework is designed to allow teachers to put in any questions which may not be tested among students, item difficulty ratios of these problems are thus, not available. Therefore, what is required is a method to calculate the difficulty level of the problems which do not depend on test results.

### **Other Measures of Problem Difficulty**

As pointed out in the previous two sections, item difficulty ratio cannot be obtained before any kind of tests are taken and is thus not suitable for the present use. On the other hand, cognitive difficulty suggested by Newman, Kundert, Lane, and Bull (1988) is based on Bloom's Taxonomy and can thus be obtained by having experts rate the items. The rating of items may not be difficult for human experts, but it would be extremely difficult for machines. Besides, as Electronic Homework is currently focused on solving mathematics problems which are mostly application problems, it may not be appropriate to use Bloom's Taxonomy to measure cognitive difficulty.

There are actually other attempts to measure the difficulty of problems besides item difficulty ratio. For example, Plake, Glover, Kraft, & Dinnel (1984) used response time to determine the cognitive complexity in a test item: response time was defined as the total time which elapsed between the presentation of the item and the response (Mason, Zollman, Bramble and O'Brien, 1992). The rationale relating it to cognitive complexity may be that "generally difficult problems require more

processing steps to finish and so the more processing steps a respondent must take to answer an item, the more time the response will require" (Loftus & Loftus, 1976; Mayer, 1975). This assertion was supported by the evidence given by Mason, Zollman, Bramble and O'Brien (1992) stating that the measure of response time was found to be correlated with the item difficulty ratio ( $r=-0.61$ ,  $p<0.001$ ). Hence, in general we can say that more difficult items require more response time although the reason behind this may not be the number of steps involved.

### Complexity of Problems

Other researchers (Jerman, 1983; Lester, 1980; Silver & Thomson, 1984; Zweng, Turner, & Geraghty, 1979) tackled this difficulty problem by relating it to the complexity of problems. According to the researchers, mathematics problems are more complex and more difficult to solve when they require several steps to obtain a solution, when subgoals must be reached before a solution can be obtained, and when the problems contain numbers that are of high computational complexity. Although there is no conclusive remarks on whether this complexity of problem is related to the item difficulty ratio (Hornke & Habon, 1986; Marzano & Jesse, 1987), this measure of problem difficulty in terms of complexity may suggest a way suitable for the present study. Factors like number of steps, number of subgoals and the computational complexity claiming to affect problem complexity, are all available before an item is tested. It is thus, possible that a measure using these terms, and may be some other factors, can be developed to represent the problem difficulty which can then be used to prioritise the problems in Electronic Homework.



## **Problem Complexity Level**

We now have three different measures of problem difficulty: the item difficulty ratio, response time and problem complexity. While response time was found to correlate with item difficulty ratio, there is no evidence reported on the relation between problem complexity and the other two measures. As the aim of studying problem difficulty is to find a suitable means or arranging problems so as to maximize the learning effect, the focus should thus, be on finding the measure that can produce the best effect rather than on the difficulties of problems. Furthermore, as both item difficulty ratio and response time would require testing on students, these measures are not suitable for the present use. Thus, the only measure for item order should be problem complexity although it is not yet defined. It is necessary for the present study to explore ways of defining problem complexity in terms of such factors as number of steps and numerical complexity, and to see whether this defined measure can be used to order problems to achieve maximum learning effects. Procedures in developing such measure will be discussed in Chapter 7.

## **Individual Differences**

Empirical findings show that not all students can benefit from traditional computer assisted instructions (e.g., Liu, 1992, Johnson, Cox, & Watson, 1994). One reason for this is that some computer systems are not well designed, resulting in poor outcomes. On the other hand, some students adapt easily to the computer environment, enabling them to derive greater benefit from computer assisted instruction systems. The exact reason for this has to be investigated. But what



concerns us is that the same may happen to the use of intelligent tutoring systems. Therefore, besides testing the effect of using Electronic Homework, it would also be necessary to find out what kinds of students are more suited to use it. The reason for doing this is obvious: if factors influencing the effects of tutoring systems can be identified, it may then be possible, first, to make the system more suitable for those who originally cannot benefit from it, and second, to maintain the learning effects of those who originally did well in the system.

As regards the characteristics that affect the use of traditional systems, Marton and Saljo (1976) argued that students can be divided into two types according to their learning styles, viz. the surface and the deep learning approaches. Biggs (1992) further divided three common ways of going about learning. According to him, learning approaches refer to predispositions to adopt particular processes, i.e. how the students usually go about learning. The three approaches are, firstly, a surface approach, which involves the “reproduction of sufficient detail to meet demands minimally”. Secondly, a deep approach which involves “understanding and coming to grips with the heart of the problem”. Finally, there is an achieving approach which “organizes oneself most cost-effectively in order to maximize the marks awarded” (Biggs, 1992). How these approaches are developed is not the concern of the present study; however, which of these approaches are used by students might possibly have some effect on the outcome of Electronic Homework or other computer assisted instructional systems. The major tutoring strategy used in the present computer system is inducing Conceptual Dissonance (CD) which requires students to link up their prior knowledge with that to be learned. For students adopting the surface approach, simply to get the task out of the way and avoid detailed resource and strategy planning, monitoring, and in depth

involvement with the task, it would be quite likely that the Conceptual Dissonance method would be useless and thus reducing the effect of the system. In fact, investigations by Jones and Kember (1994) into the effect of using a traditional computer assisted instructional program in biology showed that those with a propensity to employ a deep approach to learning are more receptive to the introduction of self-learning packages than those with a propensity towards a surface approach. It is thus possible that those who are more receptive to self-learning packages would benefit more from intelligent computer-assisted instructional systems.

While learning styles may affect how well students can adapt to the use of computer systems, their academic abilities may affect the result of the system especially as students' prior knowledge is required in producing conceptual dissonance. Effects of using Electronic Homework will be then contrasted between students with high and low academic abilities and among the three different learning approaches.

## ***CHAPTER 5***

# **HOW TO IMPLEMENT AND EVALUATE THE SYSTEM**

### **Overview**

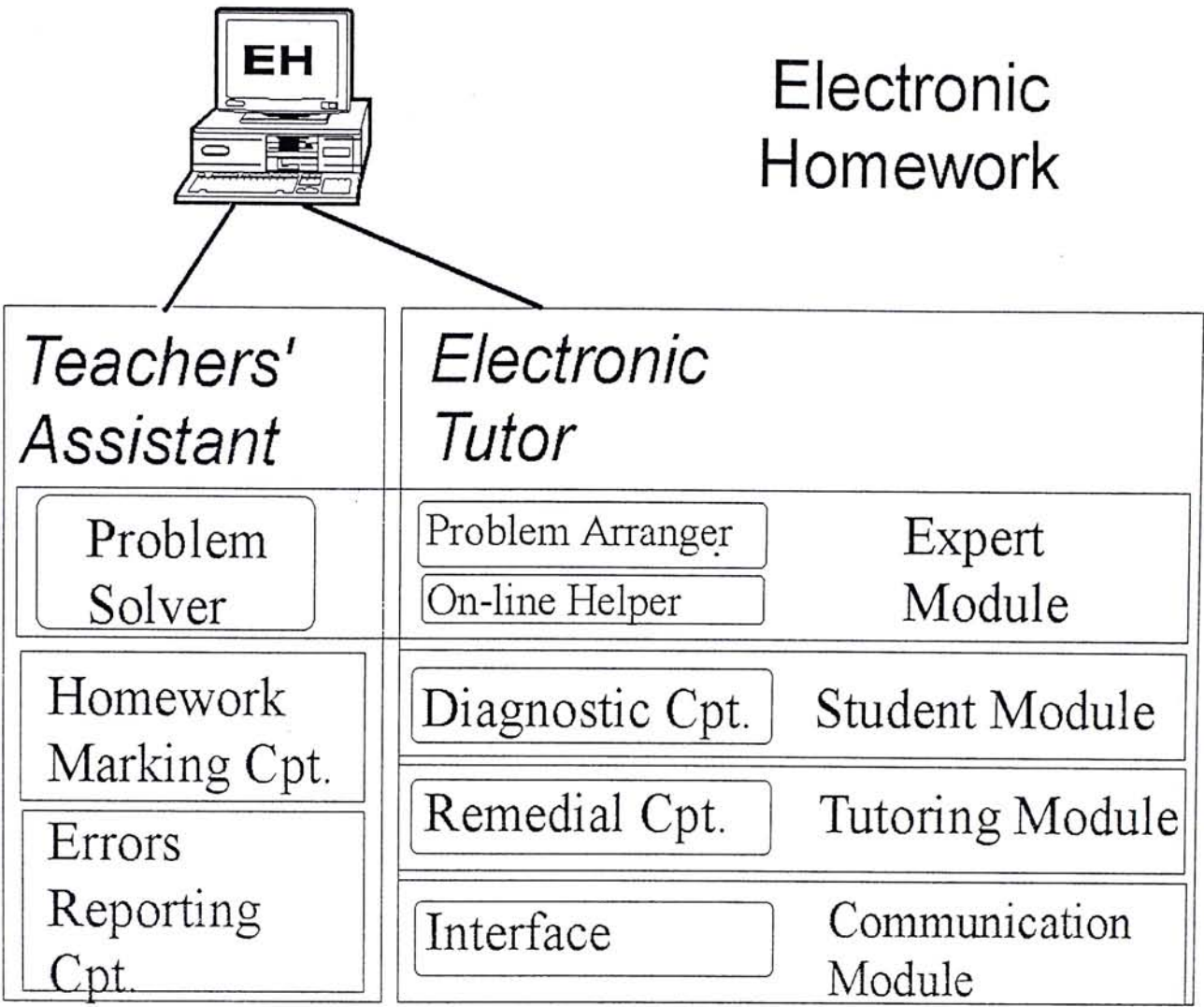
This chapter discusses how Electronic Homework was implemented and evaluated. The implementation began by building a prototype for the testing of the technical plausibility of such intelligent tutoring systems in helping students to do their homework, then followed by collecting of the knowledge required for the four modules: the expertise, student, tutoring and communication modules. The knowledge collected was incorporated into the system which then was put into real use to test its effectiveness. Different procedures and methods were used to collect the required knowledge. On evaluating the effectiveness of the system, tests on individual differences, teachers' and students' perception and students' academic abilities were also collected. Table 5.1 shows the time frame for the implementation and evaluation process and Figure 5.1 shows the structure of Electronic Homework.



Table 5.1  
Time Frame for the Implementation and Evaluation Process

Time	Stage	Expert module	Student Module	Tutoring Module			Communi- cation Module
				Tutoring Rules	Tutoring Strategies	Problem Complexity	
Jan. 1991	Proto- type	Domain Know. obtained from text books	Mal-rules simulated	Tutoring Rules simulated			Text-based interface developing
June 1992	Evaluati ng Proto- type						
June 1994					<sup>a</sup> Pretest and experi- ment Retention test and analyzing		
Sept. 1994					Strategies summar- ized		
Sept. 1995	Know- ledge acqui- sition	Domain Knowledge enriching	<sup>b</sup> Mal-rule collecting			<sup>c</sup> Students' Estimation of problem difficulty and Item Diff. Ratio collecting	<sup>d</sup> Windows-based Interface developing
Oct. 1995	Implem- -enta- tion					Teachers' estimation and criteria collecting	
Nov. 1995				<sup>e</sup> Tutoring rules collect- ing		Analyzing; Problem Complexity defined	
Feb. 1996							
Mar. 1996		<sup>f</sup> Formative evaluating Summative evaluating					
Apr. 1996							

Note. a: Page 151; b: Page 144; c: Page 156; d: Page 162; e: Page 155; f: Page 163



**Figure 5.1** Structure of Electronic Homework

## Prototype Developing

The development of Electronic Homework started by building a prototype in January 1991. The purpose of the prototype was two-fold. The first was to study the feasibility of using artificial intelligence computer language like Prolog or Lisp<sup>48</sup> to represent the knowledge required in solving problems in students' homework while the second tested whether personal computers could be used to handle such complex tasks.

The language Prolog was chosen for developing the system because it uses English-like<sup>49</sup> syntax and is thus easier to handle. It was decided that as knowledge for solving logarithm problems (Knowledge in Expert module) could be easily collected from text books, it would be incorporated first. For the student module, only typical errors were placed at this stage. Similarly, only simple tutoring strategies like giving brief feedback were used. Interface between students and the computer was also rather simple: mathematical expressions like square of 5 could only be expressed in the form of  $5^2$ . This kind of expression was not commonly used by students and thus was not adequate for long term use. However, as this was only an experimentation stage and the hardware available at that time was much less advanced than it is now, this kind of interface was considered acceptable<sup>50</sup>.

---

<sup>48</sup> Both Prolog and Lisp are called artificial intelligence languages, while Lisp is the short name of List Processing, Prolog is the short name of Programming in Logic. Both languages are suitable for logic reasoning. Refer to page 14 for detailed description.

<sup>49</sup> For example, The sentence "John is a boy" when represented in Prolog, is a statement `isa(john,boy)`. See also page 12 for earlier example.

<sup>50</sup> At that time in Hong Kong, commonly used personal computers were those using 80286 processors.



## Knowledge Acquisition

The formal procedure in developing Electronic Homework started with the acquisition of the required knowledge into the system. Knowledge acquisition was carried out around two test papers on logarithms called the Mal-rule Collecting Test (Appendix A). The first test consisted of twenty items on simplifying logarithmic expressions and the second test consisted of twelve items with half on simplifying logarithmic expressions and half on solving logarithmic equations. Items used were selected from common text books following the following criteria:

1. Types of items selected should be wide enough to cover most of the common exercises.
2. Items too complicated to be handled either by humans or by computers were not included provided that steps similar to the solving of such problems could be found in other simpler items. For example, equations like  $\log(x^3 + 6x^2 + 11x + 6) = 1$  and expressions like  $\frac{\log 2x + \log 4x + \log 5x}{\log 3x + \log 7x + \log 9x}$  might be too complex to be solved or simplified but the knowledge required might be found in other simpler problems.
3. Some items like logarithms in bases other than ten were not included to limit the scope of the system in order to make it easier to develop the system in the initial stages. In the long run, these types of problems should also be included.

The problems in these two tests, in a way, defined the knowledge domain for the present system since all the knowledge incorporated into the system at the initial stage was aimed to enable the system to solve problems or to remedy errors found in

these two tests. Basically, these two tests cover a large part of the exercises that students would have to do in their logarithms homework. Thus it was enough for the system to solve these problems at the initial stage. It is only at later stages when the system is required to solve other problems that additional knowledge will be added.

As an intelligent tutoring system has four modules, viz. the expert module, the student module, the tutoring module and the communication module, in an intelligent tutoring system, different modules would require different types of knowledge that could be acquired through different means. The following sections describe ways of acquiring the knowledge:

### Expert Module

The expert Module stores the knowledge required to complete the targeted problems in two mal-rule collecting tests. The knowledge required includes both strategic and axiomatic rules to solve the problems. Although axiomatic and strategic knowledge correspond separately to behavior according to mathematics axioms and the strategies used under different situations, both could be acquired through materials in text books. Mathematics knowledge is then coded in the form of strategic and axiomatic rules and stored into the system until all items in the two test papers can be solved.

## Student Module

As the bug identification method<sup>51</sup> was employed in the present system, a major task in developing the student module was maintaining a catalog of possible errors. In other words, the system would keep a set of mal-rules<sup>52</sup> obtained from errors made by the students. The finding of mal-rules served two purposes. Firstly, this set of mal-rules acted as bases for the system to understand students' errors. Secondly, the mal-rules collected can be a source for the investigation of why some errors happen. No tutor has access to the complete state of knowledge (Douglas, 1991). There are quite a number of mal-rules that happen infrequently, making it unlikely and impractical for a computer system to incorporate all the possible rules. If it is possible for the system to possess some kind of meta-knowledge<sup>53</sup> explaining why some of the errors happen, the mal-rules, both frequently or infrequently happened ones, can be generated by using these meta-rules (rules representing the meta-knowledge). The system could then explain more errors with fewer rules.

The procedure in acquiring knowledge for the student module thus consisted of two parts: collecting mal-rules from students and analysing mal-rules collected in order to acquire the meta-rules for errors. The following describes how this was done:

---

<sup>51</sup> The student model that keeps a catalog of bugs (errors) in addition to the correct knowledge. Refer to page 80 in chapter 3.

<sup>52</sup> Rules representing pieces of incorrect knowledge.

<sup>53</sup> Knowledge on how and when to use other knowledge. For example, to solve an equation requires the knowledge of grouping like terms and move variables to one side of the equation and constants to the other. Knowledge on how and when to use these knowledge pieces is called meta-knowledge.



### Collecting of Data

There were two test papers used to collect students' mal-rules as shown in Appendix A. The two papers contained items similar to those frequently found in common text books with paper 1 concentrating on simplifying expressions containing logarithms of numbers and paper 2 concentrating on both simplifying expressions containing logarithms of variables and solving logarithmic equations. In terms of problem complexity<sup>54</sup> and degree of abstraction, items in paper 1 are, on average easier than those in test 2 although both tests contain easy items as well as hard ones<sup>55</sup>. The exact difficulty levels of the items were determined by analysing the data obtained.

One hundred and twenty-five secondary four students from two subsidized schools of Hong Kong were invited to participate in the test programme in October 1995. As the aim of these tests was to collect mal-rules, these students were recommended by teachers as most likely to make errors. It was stipulated that these students should have medium-abilities in mathematics, the reason being that the good students would not make mistakes and poor students with insufficient knowledge to continue would soon give up and thus produce no mal-rules.

Students from the first sample school were considered to be of average ability. The best class (one in the science stream<sup>56</sup>) and an average one (one in the commercial

---

<sup>54</sup> Described in page 133 on chapter 4.

<sup>55</sup> For example, in test 1, there is a problem "simplify the expression  $\log 0.6$ ". Also, in test 2, an equation " $\log(x+6)=1$ " can be found. The solutions of both problems are both quite easy.

<sup>56</sup> When Students in Hong Kong reach secondary 4 (grade 10), they have to choose among three streams (for some schools, there are two streams) to further their studies. The three streams are: science which offers Physics, Chemistry and Biology; Arts which traditionally offers such subjects as History and Literature, and Commercial stream which offers subjects such as typing, book-keeping. Theoretically, students are not streamed according

stream) were recommended by the their teachers. Although the two classes were of different academic abilities, both were expected to have difficulty in solving the problems.

The second school was considered to be of a higher standard than the first. Hence, the recommended class were of a slightly higher than average academic ability. All three classes selected were expected to produce the mal-rules required in the present study.

The mal-rule collecting tests were administered in September 1995. Students sat the tests during their normal lessons<sup>57</sup> and were invigilated by their own teachers. Tests were administered in two separate periods and students were provided with enough time to finish the problems.

### Analysis of Data

Students' test papers in the two mal-rule collecting tests were analyzed to obtain the mal-rules. As the focus of the present study was on logarithms and it was assumed that students should be quite familiar with basic arithmetic skills such as multiplication, division and exponential, such errors were broadly categorized as groups and one mal-rule is used to represent the whole group of errors. An example is errors such as expressing the number 125 as  $5^5$  and expressing the number 8 as  $2^4$  were grouped together and represented as one mal-rule. The following is an example:

---

to their academic abilities, but is generally acknowledged that science students are more academically able than those in other streams, particularly where mathematics is concerned.

<sup>57</sup> The exact dates and time of the tests were decided by the teacher concerned.

$$[R51] \quad N(= a^n) \rightarrow a^m \text{ where } m \neq n$$

This broad categorization was not an accurate representation of errors since errors in identical forms might not be caused by the same reasons. For example, expressing the number 125 as  $5^5$  might be a careless slip since  $5^3$  was actually intended. On the other hand, expressing the number 8 as  $2^4$  might be due to the misunderstanding that the product of 4 2's is equivalent to  $2^4$ . However, if such errors were placed into different groups according to their reasons behind them, only a few errors would fall into each of these groups. As it would be very inefficient for a computer system to take care of all these infrequent mal-rules, the above categorizing means would enable the system to work more efficiently. However, the above measure was not applicable to all errors that were not related to logarithms. Errors that occurred frequently or errors that were caused by the same reasons were represented as separate rules according to the reasoning behind them. Examples of these are:

$$[R52] \quad (A + B)^2 \rightarrow A^2 + B^2$$

which represents errors such as expressing  $(x + 2)^2$  as  $x^2 + 2^2$ ;  $(3 + 4)^2$  as  $3^2 + 4^2$  and others, and

$$[R53] \quad \frac{A + B}{A + C} \rightarrow \frac{B}{C}$$

which represents errors such as expressing  $\frac{1+x}{1+y}$  as  $\frac{x}{y}$ ;  $\frac{x+2}{x+3}$  as  $\frac{2}{3}$  and others.

Besides the above classification of errors, the analysis was based on the following principles:



1. Grain size of errors or mal-rules. It would perhaps be true that every rule or mal-rule could be further decomposed into still lower level rules. However there should be a limit to the lowest level of rules, otherwise there would be no end to this decomposition process. For example, when  $\log(x + 3)$  was expressed by a student as  $\log x + \log 3$ , there could be two ways to represent this error. The first would be to simplify representing the error in terms of the abstraction<sup>58</sup> of the original expression and the newly entered expression as in [R54]:

$$[R54] \quad \log(A + B) \rightarrow \log A + \log B$$

where  $A, B$  represented any two quantities. Alternatively, the error could be interpreted as the student treated  $\log(x + 3)$  as a variable "log" times the expression " $(x + 3)$ ", or  $\log \times (x + 3)$ , then changed  $\log \times (x + 3)$  to  $\log \times x + \log \times 3$ , and finally treated  $\log \times x + \log \times 3$  as  $\log x + \log 3$ . In terms of rules, the error could be represented by the following three rules:

$$[R55] \quad \log(A + B) \rightarrow \log \times (A + B)$$

$$[R56] \quad C \times (A + B) \rightarrow C \times A + C \times B$$

$$[R57] \quad \log \times A + \log \times B \rightarrow \log A + \log B$$

where  $A, B, C$  were any three quantities. One could further decompose these rules to even lower level rules. For example, one could further investigate the reason why students would acquire the rule [R55] and express it in terms of these knowledge pieces. However, there must be an end to this process otherwise it

---

<sup>58</sup> i.e., use symbols to represent variables or constants.

would continue indefinitely. In the present study, this problem was solved by representing observed errors only. In the above example, [R54] was considered as representing an observed error since both sides of the rule could be observed directly from students' processing steps<sup>59</sup>. On the other hand, the knowledge pieces represented by the rules [R55], [R56] and [R57] might only be hypothesized based on the observed error represented by the rule [R54]. These rules might not be representing what really happened in the students' minds. Hence, unless evidence could be found showing that certain processing steps really happened, either in the form of external representation or through the protocol analyses of students' thinking processes, such steps would not be represented as rules in Electronic Homework. These lowest level rules were referred to as grains, and their corresponding sizes were called grain sizes<sup>60</sup>. Rules in the grain sizes are hereafter called the "simple rules". In particular, those representing errors were referred to as "simple mal-rules".

2. Some errors might be composite and thus, should be decomposed in terms of simple mal-rules. For example, if  $\log \frac{x+2}{3}$  was found expressed as  $\frac{\log x + \log 2}{\log 3}$ , then it would be treated as being composed of two errors of expression  $\log \frac{x+2}{3}$  as  $\frac{\log(x+2)}{\log(3)}$  and expressing  $\log(x+2)$  as  $\log x + \log 2$ . The error would then be considered as being composed of two mal-rules:

---

<sup>59</sup> If  $\log(x+3)$  expressed as  $\log x + \log 3$  is the first step of the problem solving process, then  $\log(x+3)$  is not given by the student, but it is still observable from the problem expression.

<sup>60</sup> The term grain size is generally referred to as how large the grain is. In this case, it is roughly related to how much knowledge is incorporated in the grain..

$$[R54] \quad \log(A + B) \rightarrow \log A + \log B$$

$$[R58] \quad \log \frac{A}{B} \rightarrow \frac{\log A}{\log B}$$

where again  $A$ ,  $B$  represented any quantities. It should be noted that the rules [R54] and [R58] were considered as simple mal-rule in the sense that they could both be observed from either students' work or at least the students' protocol analyses. Only those errors found expressible in terms of simple mal-rules were considered as composed errors. If no simple mal-rules could be found, the error could only be represented as a simple mal-rule like the other observed errors. No consideration would be given to how complex the problem expression was in this aspect.

### Coding of Mal-rules

Mal-rules collected were coded as clauses<sup>61</sup> by using the computer language Prolog and incorporated into the system. To enable further analysis, these rules were then grouped in terms of their nature and causes. For example, errors like those represented by rule [R55] above could be placed into the same group since they were thought to be caused by misinterpreting of the expression " $\log A$ " as " $\log$  times  $A$ " for any expression  $A$ . Detailed descriptions of these groups may be found in Appendix F, and will be discussed in later sections<sup>62</sup>. The grouping of these mal-rules enabled further analyses on these rules to be done to obtain the meta-rules<sup>63</sup>. To verify the existence of knowledge pieces described by the obtained meta-rules, six students were

---

<sup>61</sup> Basic units in the computer language Prolog. A clause is in the form of: head :- body, where head corresponds to the action and body corresponds to the condition. For example, the clause " $\text{even\_number}(X) \text{ :- divisible}(X,2)$ " means that if  $X$  is divisible by 2 then  $X$  is an even number.

<sup>62</sup> Page 178.



selected among those who participated in the mal-rule collecting tests for interview to further investigate the causes of errors. The selection of these students was based on two criteria, viz. whether evidence could be found from their test scripts in that they might possibly have made the errors described by the meta-rules and that students should have made quite a number of errors<sup>64</sup> which were not occasional ones and were thus easier to identify.

Meta-rules obtained by analyzing the mal-rules were also incorporated into the system. Besides, these meta-rules were used together with the correct rules in the system to simulate the mal-rules previously obtained. The purpose was to see whether these meta-rules could really explain the mal-rules and the process was completed by computer. Mal-rules obtained and the results of the simulation could be found in Chapter 6.

### Tutoring Module

Tutoring knowledge includes diagnostic and prescriptive knowledge. Diagnostic knowledge is that required to identify students' errors and is mainly obtained from students' errors as described above. Prescriptive knowledge involves the knowledge on when, how and what to tutor. On the question of when to tutor, discussions in previous chapters have suggested that the model-tracing approach<sup>65</sup> in which students are tutored at each step in the problem solving process. Questions left

---

<sup>63</sup> Rules that generate other rules. For more detailed discussion, refer to page 95 of Chapter 3.

<sup>64</sup> Not necessarily those with lowest scores, because some of these students might have done nothing and therefore nothing could be investigated from their scripts.

<sup>65</sup> Page 112 in Chapter 3.

then are how and what to tutor the students. Procedures to obtain the required knowledge are described separately as follows:

### **Tutoring Strategies -- How to Help Students**

Discussions in Chapter 3<sup>66</sup> showed that three different tutoring strategies may be useful in helping students to correct their errors. The strategies are called Reteaching, Model-based Remediation and Conceptual Dissonance. Reteaching is the method that just reteaches the part in which students' errors are found. In model-based remediation, a model of the student's knowledge is inferred from the interaction between the tutor and the student, and the remediation given is based on the model. In Conceptual Dissonance, students are induced to generate dissonance between the error and their previous correct knowledge, so that they can understand the reasons for their errors and thus may make fewer errors of the same nature in the future. The following sections describe how the test items were prepared and how these methods were administered to students.

#### **Materials**

The materials used included a 20-item pretest, a 20-item posttest, a 20-item retention test, a Reteaching tutoring script, a MBR tutoring script, a CD tutoring script, and a list of tasks for practice. All the test items were on solving linear algebraic equations in one variable. While items in both the pretest and posttest were those used in Sleeman, Kelly, Martinak, Ward, & Moore (1989) (Appendix B), the items in the retention test and in the list of practice tasks were made equivalent in form and

---

<sup>66</sup> Page 114.



difficulty to those in the pretest and posttest. Thus items in the different tests differed only in the numerical values used. Students could earn five marks for each item correctly done, making a total of one hundred marks for each of the three tests. The Reteaching and MBR tutoring script (see Appendix C) were also those used in Sleeman et al. (1989) but were modified to make them more suitable for the present use. The first modification was translating instructions into Chinese since the sample consisted of students who used Chinese in their daily communication. The second modification on the scripts was done to separate the effect caused by practice. In both the original tutoring scripts used by Sleeman, et al. (1989), the last steps were to let the students practise three additional items similar to the ones in which they made mistakes. In the scripts used in the present study, the effect due to this practice was studied separately, i.e., the two scripts Reteaching Only and MBR were now divided into four scripts, namely, Reteaching Only and Reteaching Only with practice, MBR and MBR with practice. This new division of groups enabled the detection of effects due to practices by contrasting the groups with and without practices.

Concerning the three tutoring strategies Reteaching, MBR and Cognitive Dissonance; the principle underlying the reteaching tutoring script was simple -- students were told that they had done a problem incorrectly and then they were retaught the correct procedure to solve the problem. On the other hand, the MBR script pointed out the errors that students had made, retaught the correct procedure, and finally, the CD script directed the tutor to point out the errors, induced dissonance in the students' mind and then retaught the required procedure.

In addition to the scripts, there was also a set of manipulative rules (Appendix D) and a set of remedial rules (Appendix E) used in the tutoring scripts. Manipulative



rules described the correct procedures for performing the tasks and were used in all the three scripts. The remedial rules, used the subjects' previous knowledge, explained why they were wrong and how to correct their errors. These rules were used in the CD script only.

### **Procedure**

The experiment was performed in four stages, viz. pretest, tutoring, posttest and retention test. One hundred and twelve secondary two students from a secondary school in Hong Kong participated in the pretest. The academic standard of the students was assessed by their teachers to be below average. Their school used Chinese as the medium of instruction. The pretest was done in June 1994. After the pretest, fifty-two students with scores of less than seventy<sup>67</sup> were identified as requiring tutoring. These students were randomly assigned to three tutoring groups: Reteaching, MBR and CD respectively. Students in the Reteaching group were just retaught the parts in which they had made errors in the pretest, while those in the MBR group were given procedurally orientated remediation of specific errors found in their solutions before reteaching. Lastly, in the CD method, errors were intentionally made dissonant to their previous conceptual knowledge. Half of the students from each group were randomly chosen to perform three more practice tasks while the other half did not. The students from the three tutoring groups were then combined as the Practice Group, and the remaining students formed the Without Practice Group.

---

<sup>67</sup> The number was chosen so that measurable number of students was selected and also the students selected would exhibit some kind of systematic error, i.e. errors not merely caused by careless work.

The tutoring was done on three consecutive days approximately three weeks after the pretest. Each student was tutored individually on the tasks in which errors were found in the pretest. The tutors were postgraduate students who were trained in a two-hour session. During the training session, the tutors were instructed to follow the tutoring scripts as shown in Appendix C and each part of the instructions was explained to them. After the training, the tutors were also required to trial-tutor on a student's working script until a consistent tutoring process among the tutors could be reached. Tutors were randomly assigned to the tutoring groups. Tutoring of each task was considered complete when the student could redo the task correctly before the tutor. For the Without Practice group, tutoring of the next task was then followed until all the tasks were dealt with. The students in the Practice group were required to do three more identical tasks before they could proceed to other tasks. The tutoring lasted for approximately 10 to 40 minutes, depending on which method was used, whether practice was required and how the student reacted.

Immediately after the tutoring, all students participated in a posttest. The tutoring was undertaken in June near the end of the 1993- 1994 academic year. After tutoring and posttest, all the subjects participated in their final examination. The retention test was taken in September 1994 when the subjects returned to school after the summer vacation. It is believed that the students were not involved in any kind of formal learning between the two posttests. The final number of subjects was fifty, as two of the participants for various reasons could not take part in the retention test. The results of the data analyses are reported in Chapter 6.



## **Tutoring Knowledge -- What to Tutor**

Although detailed results of the analyses of tutoring strategies are not discussed here, it is still necessary to report that evidence was found to show that Conceptual Dissonance produces better effects than the Reteaching and Model-based Remediation strategies. To cause conceptual dissonance to happen would require an understanding of what our students know about the topic, and the easiest way to obtain such knowledge would be to collect it from experienced teachers. By studying what experienced teachers do when they tutor students, it might be possible to obtain more insight either on how they produce conceptual dissonance or whether there could be a better strategy other than Conceptual Dissonance. The knowledge on how to help students to correct the errors was collected through questionnaires and interviews. The following paragraph describes this procedure.

Mal-rules collected through the administration of the Error-collection test papers were summarized in the form of a table as shown in Appendix F. For example, the following two rules [R59] and [R60] represented the errors most frequently found in the students' tests.

$$[R59] \quad \log(A \pm B) \rightarrow \log A \pm \log B$$

$$[R60] \quad \log A + \log B \rightarrow \log A \times \log B \text{ or } \log A - \log B \rightarrow \frac{\log A}{\log B}$$

where  $A, B$  in the above rules were any quantities. [R59] represented the error that students would express the logarithm of a sum or difference of two quantities as the sum or the difference of the logarithms of the two quantities and [R60] represented



the error that they would express the sum or difference of two logarithms as the product or quotient of the same logarithms.

Five experienced teachers of Secondary 3 to Secondary 5 mathematics were then invited to comment on the origins and ways of remediation of these errors. Their comments were then analyzed and compared in the hope of answering the following three questions:

1. Did these teachers agree on the origins of the errors?
2. Did they propose the same means to help correct students' errors?
3. Were they using the Conceptual Dissonance Strategy? If not, what did they use?

Results of the analysis are shown in Chapter 7.

### Problem Difficulty

Three measures were collected to see whether a correct prediction of problem difficulty could be obtained. The three measures were item difficulty ratio, teachers' estimation and student's estimation on problem difficulty. All these measures were collected following the Mal-rule collecting tests mentioned above. The procedures are as follows:

#### Item difficulty ratio

After the mal-rule collecting tests were administered, test papers were marked and the total number of correct responses for each question in the tests was counted and then the item difficulty ratio for each question was calculated by using this formula:

$$\text{Item difficulty ratio} = \frac{\text{Number of students who correctly answered the item}}{\text{Total number of students}}$$

One hundred and twenty-five students participated in the test and the result is shown in Chapter 7.

### **Students' Estimation on Problem Difficulty**

In the mal-rule collecting tests (Appendix A), together with each item, there is a five-point scale indicating how difficult the student thought each item was. This estimation scale ranges from very easy to very difficult and students were asked to mark on one point of them. Estimations of item difficulty of 125 students were then collected and analyzed.

### **Teachers' Estimation on Problem Difficulty**

Twenty-eight teachers completed a questionnaire containing the same items and the same estimation scale in the mal-rule collecting test papers as shown in Appendix G. Most of the teachers who participated in this test were studying for the Diploma in Education<sup>68</sup> in The Chinese University of Hong Kong. All, except one were part-time students having full-time jobs for several years. Colleagues of some of these students were also invited to participate in the test. Analyses of the results obtained will be discussed in Chapter 7.

---

<sup>68</sup> A post-graduate diploma course for training both pre-service and in-service university graduates to become qualified teachers.

## **Factors Affecting Problem Difficulty**

The above three measures on problem difficulty were for verifying that measured by the computer. To find ways for the computer to calculate this problem difficulty, knowledge on how this was done by human experts should provide valuable insight since experienced teachers practise this whenever when they assign homework to their students. Thus, the questionnaire on teachers' estimation of problem difficulty (Appendix G), not only asked teachers to predict the item difficulty, but also identify factors that they thought should be important in predicting problem difficulty. Six factors, called the difficulty factors, were assumed to affect how teachers predict the problem difficulty. They are as follows:

1. Perceived difficult steps during the problem solving process ( $f_1$ )
2. Number of steps required to finish the problem ( $f_2$ )
3. Numerical complexity ( $f_3$ )
4. Number of occurrence of "log" ( $f_4$ )
5. Number of operations in the question ( $f_5$ )
6. Degree of familiarity with the question ( $f_6$ )

Teachers were then requested to rate the importance of each of these factors on a five-point scale. Besides rating these suggested factors, the teachers were also requested to add any other factors which they thought were also important. Data obtained were then used to predict the problem complexity based on the following formula:



$$clevel = \frac{\sum_{i=1}^6 f_i r_i}{\sum_{i=1}^6 r_i}$$

where  $f_i, r_i$  are the values of the  $i$ th factors and the corresponding relative importance. Also, the values of measures of these factors were to predict teachers' estimation, students' estimation as well as the item difficulty ratio by using multiple regression. These different measures of problem difficulty will be compared in Chapter 7. However, in order to calculate the predicted problem complexity and to predict the other three measures of difficulty, measures of the six factors shown above were developed as follows:

### Perceived difficult steps during the problem solving process (f1)

This measure reflected whether the students would encounter any difficulties in the solving process. Difficult steps were assumed to be those at which students usually made errors. As errors were represented by mal-rules, the difficult steps should, therefore refer to those represented by mal-rules. Hence, difficult steps were considered as equivalent to the chances when frequently occurring errors (more than or equal to five times) would be applied. Those errors that occurred fewer than five times may have been due to careless work so they were not counted. For each question, the number of difficult steps were then countered. These are shown in Table 7.3.

### Number of steps required to finish the problem (f2)

This is defined as the number of steps that an expert would require to finish the problem. For human experts, it is possible that there may be more than one solution

path to each problem. It was therefore decided to count the number of steps of the shortest paths. Besides, there may also be discrepancies between those counted from a human expert and those from a computer system. It may not be possible to obtain both before we can choose one. Discussion on how to choose for the calculation will be given in Chapter 6. For the time being, both are listed in Table 7.3 for comparison.

### **Numerical complexity ( $f_3$ )**

A measure of numerical complexity was developed. An intuitive expression of numerical complexity would be the larger a number is, the more complex it should be. However, to avoid using too detailed a scale, which might not be necessary, the numerical complexity was measured by assigning weights to the numerical values instead of using the numerical values themselves. Every value between one and ten was assigned a weight of 1, while decimals and numerical values greater than ten were assigned weights of 2. The sum of such weights then gave the value of numerical complexity of the problem.

### **Number of occurrence of "log" ( $f_4$ )**

This is simply the number of logarithmic functions that can be found in the problem. Such numbers were counted and are listed in Table 7.3.

### **Number of operations in the question ( $f_5$ )**

This again can be obtained by simply counting the number of operations in the problem. An operation is any one of the following: addition, subtraction, multiplication, division and exponent. Results of the counting are listed in Table 7.3.

### **Degree of familiarity of the student to the question**

Students might find that some problems are more familiar than others and it is possible that they would find the familiar problems easier to solve. Concerning with the present topic, students normally learned in three stages; first the simplification of numerical expression, second the simplification of expressions involving variables, and thirdly to the solving of logarithmic functions. Further, knowledge learned at the earlier stages was also used at later stages. It is therefore reasonable to assume that problems learned at earlier stages should be more familiar to the students. This formed the basis for the value of the degree of familiarity assigned to each problem. For simplicity, all the problems on simplification of numerical expressions were assigned a value of 1, that on simplification of expressions involving variables were assigned a value of 2 and problems on solving of logarithmic equations were assigned a value of 3.

Data collected (listed in Table 7.3 of Chapter 7) were used to obtain the different measures of problem difficulty. These measures were compared and the final measure of problem complexity which was used by the computer system to predict problem difficulty was decided. How these comparisons were made and how the problem complexity was measured are shown in Chapter 7. It should be noted that factors discussed above are only assumptions. Whether they could be effective predictors of problem difficulty would be determined by the results of the analyses.



## Implementation

All the above procedures were for the purpose of developing Electronic Homework. Up to this stage, the knowledge required for the four modules: the expert, the student, the tutoring and the communication, were obtained as was the way to calculate the problem difficulty. The remaining task was to input the knowledge obtained into the computer system. While one part of the Electronic Homework, the Computer Tutor, required a lot of knowledge which had to be obtained through the procedures described above, the other part, the Teacher Assistant, mainly involved technical matters. The following sections describe how the knowledge was implemented and the Homework Assistant was developed.

### Implementation of Knowledge into Computer Tutor

Knowledge obtained for the four modules was encoded in the form of if-then rules and then translated into Prolog predicates. For example, the knowledge of how to simplify the logarithm of a product  $\log A \times B$  ( $A, B$  are any two integers) can be expressed as:

[R61] If  $\log A \times B$  is to be simplified, then express it as  $\log A + \log B$

This rule was then translated into:

[R62]  $\text{logaxiom1}(\text{log}(\text{mult}(\text{real}(\text{number1}), \text{real}(\text{number2})),$   
 $\text{plus}(\text{log}(\text{real}(\text{number1})), \text{log}(\text{real}(\text{number2}))))).$

where  $\text{logaxiom1}$  represented the meaning that this is an axiom of logarithm,  
' $\text{log}(\text{mult}(\text{real}(\text{number1}), \text{real}(\text{number2})))$ ' and ' $\text{plus}(\text{log}(\text{real}(\text{number1})),$

$\log(\text{real}(\text{number2}))$ )' representing  $\log A \times B$  and  $\log A + \log B$  respectively according to the syntax.

The knowledge acquired through the procedures described above was analyzed in the form of if-then rules, then encoded and incorporated into the system with the computer language Amzi! Prolog (Amzi, 1996). The codes can be found in one of the attached disks and details on how to access the disks can be found in Appendix S. Amzi! Prolog is mainly a text-based language which can handle inputs via the keyboard of the computer. Because students may not be familiar with typing skills, an easier input method is thus required. Another computer language Visual Basic (Microsoft, 1995) was thus used to develop an interface between the human user and the computer reasoning mechanism. The interface enabled the human user to enter expressions by just clicking the mouse at the symbols required and dragging them to the desired position. Such interface would be more user-friendly and interesting.

## Evaluation

The system was evaluated at two stages: the formative evaluation which was done at the developing stage while the summative evaluation was done when the system was completed. The following describes how these evaluations were done:

### Formative Evaluation

Two teachers and five students were invited to trial use the system in the presence of the researcher. The objective was to identify shortcomings of the system

from the users' perspective. The interviewing processes were video-taped and the analyses are at Chapter 8.

## **Summative Evaluation**

The evaluation was done after the system was developed. There were two purposes for the evaluation:

1. To check whether Electronic Homework is an effective tool for both teachers and students.
2. To investigate whether particular types of students benefit more from using this system.

## **Subjects**

Six classes of Secondary 3 students<sup>69</sup> in Hong Kong were invited to participate in the experiment. Although Electronic Homework is designed for use at home, the experiment was carried out at school during or after normal school hours. This was to avoid the contention that students of better socioeconomic backgrounds, as indicated by the possession of home computers, which have better academic results, thus causing bias in the test results. Hence only schools with computer laboratories were selected. Each class was randomly split into two groups each of which take turns to become the experimental group (Electronic Homework group) and the control group (Paper-and-pencil group). For three of the schools selected, as the number of available computers was found at the experiment stage, not enough to be used by half of the

---

<sup>69</sup>Equivalent to U.S. Grade 8.



students due to some reasons. Students without a computer to use were assigned to the control group at both sessions.

### **Data Collection**

The experiment was done in two homework sessions, referred to as Homework 1 and Homework2 respectively. Homework 1 contained items chosen from mal-rule collecting test 1 while Homework 2 contained items chosen from mal-rule collection test 2. Hence, Homework 1 was on simplifying logarithmic expressions of numerical values, while Homework 2 was on simplifying logarithmic expressions of variables and solving logarithmic equations.

Both parts of the experiment started with a presentation period during which both groups were taught by the same teacher with materials used in the homework. After that, they were given the homework with each half of the class taking turn to become the experimental and control groups.

Students in the experimental group had their homework done in the computer laboratory while the others worked in a classroom. Both groups were invigilated to avoid cheating and discussion among the students. The students in the experimental group were given a ten minutes briefing session to familiarize them with the use of a computer to do their homework. After that, they were free to do their homework on the computer assigned to them. During this period, help was given only to questions on how to use the computer program but not to questions on the subject contents. All details on number of correct items and errors made were recorded on disks for individual students. The floppy disks were collected after the experimental session.

For the control group, homework was done as usual and the students were not allowed to discuss among themselves.

Besides doing their homework, each participant was required to complete the following tests:

1. Retention Test: This is to test the quantity of learned material which remained after the tutorial. The test was done approximately one week after the homework.
2. Students' Learning approach: This is to identify whether students used a surface approach or a deep approach in learning, a sample of which can be found in Appendix H. This was done any time between the experiment and the retention test at the discretion of the teachers.
3. Perception on Electronic Homework: Students' opinions on using Electronic Homework were collected (Appendix I).

Also, students' mathematics scores in a most recent examination or test results were collected as indicators of their academic abilities. At the same time, teachers of the participant classes were requested to complete a questionnaire to collect their opinions on using Electronic Homework (Appendix J).

### **Analysis of Data:**

In order to study whether students of different academic abilities would benefit differently on the use of Electronic Homework, students were categorized into the high, median and low ability groups according to the mathematics scores collected. These mathematics scores were collected from different schools, and would thus be of different standards. They were adjusted by the students' scores obtained in their

homework, a test taken by all participants. Thus the mathematics score of each student was first translated into the standard score by finding how far the score was from the mean in the units of the class standard deviation of the mathematics score, and then this standard score was translated into the ability score by multiplying the standard score to the standard deviation of the homework scores of students in the same class. While the standard score of each student can be obtained by using the following fraction:

$$\frac{\text{student's mathematics score} - \text{class mean of mathematics score}}{\text{class standard deviation of mathematics score}}$$

The formula used to calculate the ability score was thus as follows:

$$\text{ability score} = \frac{\text{student's mathematics score} - \text{class mean of mathematics score}}{\text{class standard deviation of mathematics score}} \times \text{standard deviation of homework score in the class}$$

where ability score is the score used to allocate students into the three ability groups, the mathematics scores refer to students' results in their examination or test, and the homework score is the score obtained in doing either the Electronic Homework or conventional homework.

Students scores in the Learning Process Questionnaire were added up and converted to two decile scores: the surface score and the deep score. Those with their deep scores exceed their surface scores by 2 were classified as using deep-biased while those with surface scores exceed their deep scores by 2 were classified as the surface-biased. Finally, students were classified according to their homework types, i.e., Electronic or conventional homework.



### **Effectiveness of Electronic Homework:**

The effectiveness of the Electronic Homework was studied by contrasting the effects on the scores obtained in the homework and in the retention test between the homework method, learning approaches, abilities and levels of abstraction of problems. Multiple Analysis of Variance was employed with Scores in the homework (electronic and paper-and-pencil) and retention test as the dependent variables and Method of doing homework (Electronic Homework versus pencil and paper), academic abilities, levels of abstraction of problems and learning approaches as the independent variables. The results of analysis can be found in Chapter 8.

### **Perception of Electronic Homework**

The questionnaires on teachers' and students' perception of Electronic Homework were analyzed to reflect their opinions on using the system. The results of the analyses are shown in chapter 8.

## **Chapter Summary**

This chapter describes the procedure involved in the development and evaluation of Electronic Homework. The developing of Electronic Homework was started by building a prototype to test its feasibility and was then followed by acquiring the necessary knowledge. Different ways were used to gain the required knowledge, viz. Domain knowledge, which is required to solve the problems, was acquired through text books and experts; students errors were obtained through tests; tutoring knowledge including the tutoring strategies and the knowledge required to produce conceptual dissonance were obtained by tests as well as experimentation.

The knowledge was acquired through tests and interviews. This was then implemented into the system in the computer language Prolog. While Prolog could not offer convenient input methods for the students, another computer language called Visual Basic was used to build up a more user-friendly interface so that students could enter their expressions easier. Finally, the system was evaluated on whether it can be of help to students in general and what types of students would be beneficial from using the system.

## **CHAPTER 6**

# **KNOWLEDGE ACQUIRED**

### **Overview**

The knowledge required in Electronic Homework was acquired by the procedures described in the last chapter. This chapter reports on the results of the acquisition processes. While the knowledge required for the expert module was relatively easy to obtain, knowledge for the student module required a great deal of effort both in collecting and analyzing. The result was a catalog of mal-rules and a set of meta-rules that could be used to explain the formation of the mal-rules. From the meta-rules deduced, it is then argued that besides the traditionally believed causes of errors such as impasse repairing and misgeneralization, misperception should also be considered a major source of errors and this has never been reported.

Also in this chapter is a report on the results of the experiment carried out on the tutorial module which compared the three different tutoring strategies: the Conceptual Dissonance method<sup>70</sup>, Reteaching Only and the Model-based

---

<sup>70</sup>The method of contrasting students' errors with their previous knowledge. Refer to page 117 of chapter 3 for more details.



Remediation<sup>71</sup>. The Conceptual Dissonance method was found to produce the best effect among the three. Teachers' opinions collected on the causes of errors and the ways to correct them were then contrasted with the meta-rules obtained for the student module and the tutoring strategies obtained from the above experiment. A final discussion will be on what is to be placed into the tutorial module of the system.

## Expert Module

This module contains the knowledge required to solve the problems in the desired domain in the manner of an expert. For the time being, Electronic Homework is aimed at helping students to simplify logarithmic expressions and solve logarithmic equations. It is thus necessary for the system to possess the required knowledge. Discussion in Chapter 3<sup>72</sup> showed that two types of rules, the strategic and axiomatic rules, should be used to represent the knowledge. The rules were acquired from text books and experienced teachers and coded in the form of clauses<sup>73</sup>. A total of 555 strategic rules and 64 axiomatic rules were thus obtained. Examples of such rules are as follows:

---

<sup>71</sup>The method of forming a model of the student based on what observed and then remedying. Refer to page 115 of chapter 3 for more details.

<sup>72</sup>Page 72.

<sup>73</sup>A basic unit in the computer language Prolog. A clause is in the form of: head :- body, where head corresponds to the action and body corresponds to the condition. For example, the clause "even\_number(X) :- divisible(X,2)" means that if X is divisible by 2 then X is an even number.

[SR1]

strat(*Rulename*,log(mult(real(*X*),real(*Y*))),*Newexp*,1,0):-

axiomlog(*Rulename*,log(mult(real(*X*),real(*Y*))),*Newexp*).

[AR1]

axiomlog(logprop1,log(mult(real(*X*),real(*Y*))),plus(log(real(*X*)),log(real(*Y*)))).

Expressions in italics in the above rules were variables and “log(mult(real(*X*),real(*Y*)))” and “plus(log(real(*X*)),log(real(*Y*)))” are the computer codes of the expressions “log( $X \times Y$ )” and “log(*X*)+log(*Y*)” respectively.

When expressed in terms of common language, [SR1] is in the form of:

IF                      The task is to simplify an expression “log(mult(real(*X*),real(*Y*)))”,

AND                  There is an axiom simplifying “log(mult(real(*X*),real(*Y*)))” into a  
new expression *Newexp*,

THEN                  Return *Newexp* as the answer and *Rulename* as the name of the  
axiom;

AND                  Marked the Number of steps required as 1 and Truthness as True (0).

For the axiomatic rule [AR1], it simply means that:

log(mult(real(*X*),real(*Y*)))=plus(log(real(*X*)),log(real(*Y*)))

and the name of the rule is “logprop1”.

All the strategic rules and axiomatic rules can be found in one of attached disks (Refer to Appendix S on details on how to access the disks). Besides these rules, there

were actually other rules in the system that took care of things like simple arithmetic and recognizing equations<sup>74</sup>. A simple example is as follows:

$$\text{reduce}(\text{Exp1}, \text{Exp2})$$

which is a rule to simplify the expression Exp1 and return Exp2 as the answer. For example, if  $\text{Exp1} = \text{plus}(\text{real}(2), \text{real}(3))$  then Exp2 will be given as  $\text{real}(5)$ . Note that  $\text{plus}(\text{real}(2), \text{real}(3))$  is the computer code for “2+3” and  $\text{real}(5)$  is the computer code for “5”

The rules were incorporated into the system and it was found that the system could solve problems similar to those used in the mal-rule collecting tests, provided the numerical values were not too complex.

## Student module

The knowledge for the student module includes the mal-rules as well as meta-rules deduced from the mal-rules. Results of the analyses are as follows:

### Mal-rules

According to the analysis<sup>75</sup> shown in Chapter 5, 114 mal-rules were obtained (Appendix F), which were then encoded into the computer system. To facilitate further analysis, the mal-rules were grouped according to the nature of the errors. Some of the groups were further divided into subgroups. Table 6.1 shows the names of

---

<sup>74</sup> It is easy for human beings to recognise an equation but it is not so simple for machines. The system has to include quite a number of computer codes to enable the system to dissect and recognise equations and other functions. These codes are not included here but may be found the listing of the whole system in Appendix S.

<sup>75</sup> Page 145.



the groups and subgroups and the corresponding number of mal-rules found in the groups.

As seen in Table 6.1, the mal-rules are divided into five groups. The first two groups are related to logarithms, one on simplifying logarithmic expressions and the other on solving logarithmic equations. The next two groups are mal-rules about other algebraic manipulations, one for solving equations and the other for simplifying expressions. Finally, the last group contains the rules representing the errors called slips which are possibly caused by careless work.

Table 6.1

Table showing groups of mal-rules obtained

<i>Groups</i>	<i>Subgroups</i>	<i>No. of Occurrences</i>	<i>Percentage</i>
<i>Concerning simplifications of logarithmic expressions</i>	<i>Because log A is treated as of log times A</i>	<i>164</i>	<i>14.92%</i>
	<i>Due to incorrect distributive law and log A as log times A</i>	<i>160</i>	<i>14.56%</i>
	<i>Due to confusion caused by the logarithm axioms</i>	<i>281</i>	<i>25.57%</i>
<i>Concerning solving logarithmic equations</i>	<i>Unable to reject roots that cause logarithms of negative numbers</i>	<i>11</i>	<i>1.00%</i>
<i>Concerning solving equations without the other side of the logarithms</i>	<i>Related to moving a term to without the other side of the equation</i>	<i>21</i>	<i>1.91%</i>
	<i>Related to roots of a number</i>	<i>45</i>	<i>4.09%</i>
	<i>Related to multiplying the whole equation</i>	<i>1</i>	<i>0.09%</i>
<i>Algebraic Manipulation (other than that related</i>	<i>mix up * with +, / with -</i>	<i>17</i>	<i>1.55%</i>

## Knowledge Acquired

to properties of  
logarithms)

Neglecting do mult. or div. first and other orders such as that in fractions	17	1.55%
Concerning Distributive Law	180	16.38%
About Index	35	3.18%
Other Manipulation Errors	65	5.91%
Slips	39	3.55%
Unclassified <sup>a</sup>	63	5.73%

---

Note: <sup>a</sup>: Complex errors that are not classified due to the fact that there may be different possible explanations.



## **Strategies for Handling Mal-rules**

As pointed out by Douglas (1991), no tutor has access to the complete state of knowledge of the student. Mal-rules in the present study were obtained from 125 students - a very small number when compared to the whole population of secondary school students. The mal-rules collected cannot, therefore, be said to be exhaustive unless the whole population is tested. While the testing of the whole population is practically impossible, it is still possible to pool the knowledge of as many teachers as possible so that the computer system can understand more errors than any single teacher. This idea of accumulating knowledge may be a possible way in the future but may not be practical for the time being, since our students are still using rather low-end computers which may response very slowly when a large number of mal-rules have to be scanned.

On the other hand, it may not be necessary for a computer system to incorporate all the mal-rules to diagnose students' errors since human tutors themselves also cannot possibly possess all the mal-rules. Human tutors would possibly have knowledge of errors that occurred frequently, but for those which occurred relatively infrequently, they would sometimes rely on their background knowledge and experience to try to understand the reasons for the errors. Remediations were made on these inferred reasons of the errors. If the computer system could be taught to understand students' errors, it would be easier for the system to handle new errors. Besides, the size of the system could be much reduced since there will be less need to incorporate relatively infrequent errors.

## Understanding the Errors

Representing errors with mal-rules is a way of understanding errors. In previous sections<sup>76</sup>, it was claimed that intelligent tutoring systems were superior to traditional computer-assisted instruction systems since the former do not have to prewire all the possible errors in the system. Intelligent tutoring systems achieve this by using mal-rules, each of which represents a large number of similar errors. For example, a mal-rule in the form:

$$\log(A + B) \rightarrow \log A + \log B$$

represents the whole set of errors in the form of:

$$\log(1 + 2) \rightarrow \log 1 + \log 2; \log(1 + 3) \rightarrow \log 1 + \log 3; \dots; \log(2 + 1) \rightarrow \log 2 + \log 1; \dots$$

A rule thus represents thousands of instances and this abstraction process of representing many instances as one rule should therefore be a convenient and economical way to store knowledge in any computer systems and even in human beings. If this is an economical way of storing knowledge and we know that human beings do store knowledge in this way, then there should be no reason why this abstraction process cannot be extended to include a higher level of meta-rules, each of which represents numbers of rules. Data collected in the present study showed such possibilities.

Among the six groups of mal-rules in Table 6.1, it can be seen that the most frequent errors are in the first group and together they made up 45.05% of the total number of errors. The other major group of errors came from that on algebraic

---

<sup>76</sup> Page 17.



manipulation, another group was related to properties of logarithms (28.57%) and a large proportion of errors came from that concerning the mis-use of distributive laws (16.38%). Besides these three groups, the other errors were found to be relatively rare. It may be necessary for us to differentiate between a frequently occurring error (e.g. one which happened 50 times in the mal-rule collecting tests) from a highly probable error (e.g. one which happened 5 out of 7 chances). Since if the first error had a chance of 1000 times to occur while the second one only had 7 times, then although the second one occurred less, it might be a more important error to study. However, in the case of the data obtained here, as the occurrence of errors in the most frequent groups is distinguishably higher than the others, and the infrequent errors did not lack opportunities to occur, (for example, there are quite a lot of chances in the tests to make errors such as  $2 \times 3 = 7$ ;  $2 \times 3 = 8$ ; and so on, but few such errors occurred), it was thus possible that the most frequent and the most probable errors were those concerning logarithms and those concerning distributive laws.

The first groups of errors were further divided into three subgroups, viz. those where the logarithm of an expression was treated as a quantity “log times that expression”; those which incorrectly used the distributive law in addition to the errors in the first subgroup; and those showing confusion caused by the logarithm axioms. The purpose of this division was to find a meta-rule for each of the subgroups so that each mal-rule in a subgroup can be explained as being composed of the meta-rule and the correct rules. It is in this case that the system can be said to understand students’ errors and that the rules representing infrequent errors can be deleted from the system, only to be generated when they are needed. Systems equipped with meta-rules are



thus representing more knowledge pieces with fewer rules incorporated. The following sections describe how these could be done.

### **Errors caused by “ $\log A$ ” treated as “ $\log$ times $A$ ”**

Mal-rules in this subgroup are in many ways similar to some correct rules. For example, mal-rule AA1 (see Appendix F) says:

$$[\text{AA1}] \quad \log(A + B) \rightarrow \log A + \log B$$

which means the logarithm of the sum of two quantities  $A$  and  $B$  is equal to the sum of logarithm of  $A$  and the logarithm of  $B$ . In this rule, “log” is a function operator. When it is operated to a quantity  $X$ , together they become the logarithmic function of  $X$  (denoted as  $\log X$ ). However, the outlook of  $\log(A+B)$  is exactly the same as the product of quantity with the sum of two quantities such as  $K(A+B)$  ( $K$  here represents any quantity. If the  $K$  here is replaced with  $\log$ , it looks exactly the same as the left hand side of [AA1]). Hence, if a student does not know the difference between the expressions  $\log(A+B)$  and  $K(A+B)$ , or for any reason he or she perceives  $\log(A+B)$  as  $K(A+B)$ , an error in the form of [AA1] is then generated<sup>77</sup>. Hence, if rules [MR601], [MR602] are created as follows:

---

<sup>77</sup>Even experts like mathematics teachers would sometimes commit such errors. However, even though the same error is exhibited, the reasons for the error may be different. For the teachers, this may just be careless work; but for the students, these might be a sign of deficient learning.

$$[\text{MR601}] \quad \log X \rightarrow \log \times X$$

$$[\text{MR602}] \quad \log \times X \rightarrow \log X$$

which can be interpreted as the logarithm of a quantity  $X$  is treated as a quantity "log" times the quantity " $X$ ", then together with the correct distributive law:

$$[\text{R601}] \quad k(A + B) \rightarrow kA + kB$$

the mal-rule [AA1] then follows. For example, the error:

$$[\text{R602}] \quad \log (2+3) \text{ is written as } \log 2 + \log 3$$

can be explained as consisting of the following steps:

$$[\text{R603}] \quad \log (2+3) \rightarrow \log \times (2 + 3)$$

$$[\text{R604}] \quad \log \times (2 + 3) \rightarrow \log \times 2 + \log \times 3$$

$$[\text{R605}] \quad \log \times 2 + \log \times 3 \rightarrow \log 2 + \log 3.$$

While the steps [R603], [R605] are the applications of the meta-rules [MR601] and [MR602] respectively, [R604] is the application of the correct distributive law [R601].

Since the correct rule [R601] is already in the expert module, the mal-rule [AA1] is said to be explained by the addition of two meta-rules [MR601] and [MR602] into the system. Adding two extra rules to explain one mal-rule is of course no advantage to the system. However, it was found that by adding these two meta-rules to the computer system, all the mal-rules in this subgroup can be generated from the meta-rules and other related correct rules. The whole subgroup of mal-rules can now be explained by just two meta-rules.

## **Protocol Analyses**

Evidence showing that students could mix up  $\log(A+B)$  and  $K(A+B)$  can be found in the protocol analysis of student S1 (detailed analysis can be found in Appendix R). When this student was asked to solve the equation:

$$\log(x + 6) + 1 = 0$$

she responded by giving the following:

$$\log x + \log 6 = -1$$

Table 6.2 shows how this student responded to the researcher's question on reasons why she did this.

Similar dialogues could be found in the same protocol and that of other students. Hence it is quite clear that at least some students would mix up the logarithm of a quantity and the product of a quantity named "log" with the other.

## **Errors due to incorrect use of distributive law in addition to treating "log A" as "log times A"**

Before learning logarithms, the students should have learned the distributive laws:

$$[R606] \quad k \times (A + B) \rightarrow k \times A + k \times B; \quad k \times A + k \times B \rightarrow k \times (A + B)$$

$$[R607] \quad k \times (A - B) \rightarrow k \times A - k \times B; \quad k \times A - k \times B \rightarrow k \times (A - B)$$



Table 6.2

Dialogue showing that a student treated "log A" as "log times A"<sup>78</sup>

- 
- T Q: Why did you change  $\log (x+6)$  to  $\log x + \log 6$ ?
- 2 A: I just multiplied the "log" into the brackets
- 3 Q: Multiply "log" inside the bracket? Do you mean that "log" can be treated as multiplication? Do you really mean that?
- 4 A: Yes.
- 5 Q: Is "log" the same as multiplication?
- 6 A: (Repeats the question). Is "log" the same as multiplication?
- 7 Q: When you take the "log" of a number, is it identical to multiplying the number with something?
- 8 A: ...., Is "log" the same as multiplication?..I really don't know.
- 

---

<sup>78</sup> The original Chinese dialogue was translated into English for the convenience of the readers.

If the learning was not complete<sup>79</sup> enough, the two rules might be misgeneralized as:

$$[\text{MR603}] \quad k \times (A \oplus B) \rightarrow k \times A \oplus k \times B; k \times A \oplus k \times B \rightarrow k \times (A \oplus B)$$

where  $\oplus$  represents any operations between two quantities. This generalized mal-rule would eventually imply:

$$[\text{MR604}] \quad k \times (A \times B) \rightarrow k \times A \times k \times B; k \times A \times k \times B \rightarrow k \times (A \times B)$$

$$[\text{MR605}] \quad k \times \frac{A}{B} \rightarrow \frac{k \times A}{k \times B}; \frac{k \times A}{k \times B} \rightarrow k \times \frac{A}{B}$$

In addition, when “ $\log A$ ” is perceived as “ $\log$  times  $A$ ”, the following mal-rules will then be deduced:

$$[\text{MR606}] \quad \log(A \times B) \rightarrow \log A \times \log B; \log A \times \log B \rightarrow \log(A \times B)$$

$$[\text{MR607}] \quad \log \frac{A}{B} \rightarrow \frac{\log A}{\log B}; \frac{\log A}{\log B} \rightarrow \log \frac{A}{B}$$

The above two mal-rules [MR606] and [MR607] were found to describe 133 errors or 12.10 % of all that identified. By just adding the rule [MR603], the computer system could be used to simulate the above two rules as well as others in the same subgroup. Thus, all the rules in this group could be explained in terms of just one meta-rule in addition to the correct rules.

Evidence was also found from the protocol analysis that students would use the rules [MR606] and [MR607] since they might think that [MR603] is correct. The following extract from the protocol analyses of the problem solving process of student S2 in Appendix R serves to illustrate this:

---

<sup>79</sup> According to VanLehn (1990), errors are caused by incomplete learning.

Table 6.3

Dialogue showing the reason why a student treated " $\log AB$ " as " $\log A$  times  $\log B$ "

- 
- |    |   |
|----|---|
| 1  | Q: You changed $2 \log 2 + \log 5$ to $2 \log 10$ . Can you tell me why you did that?   |
| 2  | A: Mmm. This is what I usually do. When it is $\log 10$ , I would change it to $\log 2$ times $\log 5$ , so..   |
| 3  | Q: Do you mean that you have changed $2 \log 2 + \log 5$ to $2 \times \log 2 \times \log 5$ , then ....   |
| 4  | A: Take the "log" out.  |
| 5  | Q: Take the "log" out? You now have $\log (2 \times 5)$ . Right? Do you think this step is correct or not?  |
| 6  | A: I think so.  |
| 7  | Q: Yes? Is there a formula for that? Is there?  |
| 8  | A: There doesn't seem to be.  |
| 9  | Q: Think again.   |
| 10 | A: No, there isn't.   |
| 11 | Q: There seems to be no such formula. But still you think it is correct. In fact, you did not think about using any formula, you just felt that it should be correct. Am I right? |
| 12 | A: I felt that it was approximately the way it should be.   |
| 13 | Q: Yes. The way it should be.   |
-



From lines 3 to 5, it can be seen that the student did think that  $\log 2 \times \log 5$  was equal to  $\log (2 \times 5)$ . Although she could not give a reason for this, from line 4, we can see that she did think that "log" can be taken out. Again, we cannot know why she thought that "log" can be taken out, but from another extract of the same protocol as shown in Table 6.4, we can see that this student did think "log" can be treated as multiplication:

Table 6.4

An Extract of the protocol of student S2 on log A as log times A.

---

1	Q:	For $\log(x+6)$ , you changed it to $\log x + \log 6$ . Why did you do this?
2	A:	I thought the "log" could be taken out, so that I could have 2 logs.
3	Q:	Why could both these quantities have "log"?
4	A:	Because at that time, I could multiply the "log" back when I needed to.

---

From lines 2 and 4, we can see that when the student tried to take the "log" out, she was actually thinking that the logarithm of a quantity was identical to the product of a variable "log" and the quantity that followed. Combining these, we can then say that the errors represented by rules [MR606] and [MR607] may possibly be explained by the facts that students might think that logarithm is some kind of multiplication and that when a quantity is multiplied to a product or ratio, the result will be the same as if the quantity was multiplied to the factors. Besides, if students did think that logarithm is multiplication, then from line 1 and 2 of Table 6.4 we can find that the student did think that  $K \times (x+6)$  was identical to  $Kx+6K$ . Hence, this student should also possess the knowledge piece represented by [R606] and perhaps also [R607]. In other words, it might be possible that this student has already acquired the knowledge piece represented by rule [MR603] and that the error was generated by the use of this [MR603] and the correct rules [R606] and [R607].

### Errors due to confusion caused by the logarithm axioms

This subgroup consists of all the errors which cause confusion in students by the introduction of logarithms. The logarithm of a quantity has properties that distinguish it from the other calculations that students are used to and thus causes confusion among them. Examples of these properties are:

$$[R608] \log A + \log B \rightarrow \log AB \text{ and}$$

$$[R609] \log A - \log B \rightarrow \log \frac{A}{B}$$

If expressed as productions, the above axioms should be in the form of:

- P1    IF        *Expression1* plus *Expression2* is to be simplified,  
         AND     *Expression1* is a logarithmic function in the form of  $\log A$   
         AND     *Expression2* is a logarithmic function in the form of  $\log B$   
         THEN    express it as logarithm of  $A$  times  $B$
- P2    IF        *Expression1* minus *Expression2* is to be simplified,  
         AND     *Expression1* is a logarithmic function in the form of  $\log A$   
         AND     *Expression2* is a logarithmic function in the form of  $\log B$   
         THEN    express it as logarithm of  $A$  over  $B$

As learning is never complete (VanLehn, 1990), when the productions P1 and P2 were learned, students may fail to identify what should be placed into the action parts of the productions, thus productions [P3] and [P4] as shown below might be formed:

- P3    IF        *Expression1* + *Expression2* is to be simplified,  
         AND     *Expression1* is a logarithmic function  
         AND     *Expression2* is a logarithmic function  
         THEN    express it as *Expression1* times *Expression2*
- P4    IF        *Expression1* minus *Expression2* is to be simplified,  
         AND     *Expression1* is a logarithmic function  
         AND     *Expression2* is a logarithmic function  
         THEN    express it as *Expression1* over *Expression2*

When expressed in rules, these productions would be like this:



$$[\text{MR608}] \quad \log A + \log B \rightarrow \log A \times \log B$$

$$[\text{MR609}] \quad \log A - \log B \rightarrow \frac{\log A}{\log B}$$

According to the data collected in the present study, 31 errors were found to be described by these two mal-rules [MR608] and [MR609]. Together they made 2.82% of the total number of errors. If the students generalize  $\log A$  and  $\log B$  as any expression involving logarithmic functions, then the productions P3 and P4 would become:

- P5    IF        *Expression1* plus *Expression2* is to be simplified,  
          AND     *Expression1* involves logarithmic functions  
          AND     *Expression2* involves logarithmic functions  
          THEN    express it as *Expression1* times *Expression2*
- P6    IF        *Expression1* minus *Expression2* is to be simplified,  
          AND     *Expression1* involves logarithmic functions  
          AND     *Expression2* involves logarithmic functions  
          THEN    express it as *Expression1* over *Expression2*

When expressed as rules, these two productions become:

$$\begin{aligned} \text{[MR610]} \quad & \text{Expression\_in\_log1} + \text{Expression\_in\_log2} \\ & \rightarrow \text{Expression\_in\_log1} \times \text{Expression\_in\_log2} \end{aligned}$$

$$\begin{aligned} \text{[MR611]} \quad & \text{Expression\_in\_log1} - \text{Expression\_in\_log2} \\ & \rightarrow \frac{\text{Expression\_in\_log1}}{\text{Expression\_in\_log2}} \end{aligned}$$

where *Expression\_in\_log1* and *Expression\_in\_log2* are expressions involving logarithms.

By including these two rules and their converses<sup>80</sup> [MR612] and [MR613]:

$$\begin{aligned} \text{[MR612]} \quad & \text{Expression\_in\_log1} \times \text{Expression\_in\_log2} \\ & \rightarrow \text{Expression\_in\_log1} + \text{Expression\_in\_log2} \end{aligned}$$

$$\begin{aligned} \text{[MR613]} \quad & \frac{\text{Expression\_in\_log1}}{\text{Expression\_in\_log2}} \\ & \rightarrow \text{Expression\_in\_log1} - \text{Expression\_in\_log2} \end{aligned}$$

into the system, it was found that all mal-rules in the present subgroup could be predicted.

There was quite a lot of evidence showing that students were confused by the logarithm properties. Extraction of the protocols (Appendix R) of a student S3 is shown in Table 6.5 as an example:

---

<sup>80</sup>If a production is in the form of “if p then q”, then its converse will be in the form of “if q then p”, i.e., the converse is formed by interchanging the condition and action parts of a production. Note that even if a production is correct, it may be possible that its converse is wrong. Example of this is that the production “if x is an even number, then x is an integer.” is correct, but its converse “if x is an integer then x is an even number” is surely wrong. However, in the case of the mathematics axioms stated here, both the productions and their converses are correct.

Table 6.5

Extraction of Protocol of student S3 to illustrate the confusion caused by properties of logarithms.

- 
- 1 Q: Now we have  $\log 60 - \log 6$  here. You changed it to  $\frac{\log 60}{\log 6}$ . Is there a formula for that? Why did you do that?
- 2 A: Because it seems that you can always change subtraction to division.
- 3 Q: Subtraction to division?
- 4 A: Yes. That's why I changed it.
- 5 Q: So now you have  $\log 60$  minus  $\log 6$  became  $\log 60$  over  $\log 6$ . Is that right?
- 6 A: Yes.
- 7 Q: Do you mean that you can change all subtraction to division?
- 8 A: Not exactly. But I don't know why I did it that time.
- 9 Q: What made you think at that time that subtraction could be changed to division?
- 10 A: At that time, ..., I saw the subtraction and I thought there could be an integer which then became one.
- 11 Q: So..., would it be because there was a "log"? Is it true that whenever there is a "log", you can change a subtraction to division?
- 12 A: Yes, because when I looked at the other questions, I can always delete something by doing so.
-



The above example shows that at least some students did mix up subtraction and division (and also multiplication and addition as seen in other parts of the same interview and interviews with other students) due to the introduction of logarithms. The evolution of the productions from P1 to P5 and that from P2 to P6 may well explain such errors.

### Causes of Confusion

It should be noted that the process of modifying the rule [MR608] to [MR610] and that of [MR609] to [MR611] can be explained in terms of generalization since both condition and action parts of rule [MR608] are special cases of the corresponding parts of rule [MR610]<sup>81</sup>. In other words, expressions applicable to rule [MR608] such as "log 2" form a subset of the set of expressions such as "3 log 2" that are applicable to rule [MR610]. An identical relationship can also be found between the rules [MR609] and [MR611]. However, there seems to be no such relationship between the rules [R608] and [MR608], and between [R609] and [MR609]. Take the pair [R608] and [MR608] as an example:

$$[R608] \quad \log A + \log B \rightarrow \log AB$$

$$[MR608] \quad \log A + \log B \rightarrow \log A \times \log B$$

Although the condition parts of the two rules are identical, there cannot be any subset relationship between the action parts. In other words, we cannot say that  $\log 2x$  is an instance of  $\log 2 \times \log x$  or that  $\log 2 \times \log x$  is an instance of  $\log 2x$ . Neither

---

<sup>81</sup> Refer to rules [MR68] and [MR610], log A and log B in [MR68] are themselves expressions in logarithms and hence "log A + log B" is a special case of "Expression\_in\_log1 + Expression\_in\_log2" and "log A × log B" is a special case of "Expression\_in\_log1 × Expression\_in\_log2". While generalization is the process of deducing general from specific, the process of deducing [MR610] from [MR68] is thus a generalization process.

one of the two rules can be said to be the result of generalization for the other. Hence, the cause of these errors cannot be traced to misgeneralization.

On the other hand, it is not possible that these errors are caused by impasse-repairing as suggested by VanLehn (1982a, 1982b). According to VanLehn's Repair Theory, students might get stuck during a problem solving process and would do some problem solving to overcome the difficulty. The point where a student gets stuck is called an impasse and the problem solving process to overcome this impasse is called the repair. When the repairing is done incorrectly, an error occurs. Also, this impasse-repair pair might be remembered as a mal-rule and become a possible source of future errors when an identical situation is encountered.

Hence, if an error is caused by either repairing an impasse, or an impasse-repair pair that had occurred before, the necessary condition would be an impasse experienced by the student some time before the error happened. However, if we look at the courses of the generation of the mal-rules [MR608] and [MR609], it seems unlikely that such impasse does exist. Take the course for [MR609] as an example: [MR609] represents the production P4 and is thought to be a variant of the production P2, which represents [R609], due to imperfect learning<sup>82</sup>.

Hence, before a student can have the knowledge piece represented by [MR609], he or she should have, at least, come across [R609]. Now suppose this student has to simplify an expression:

$$\log 60 - \log 6$$

---

<sup>82</sup> Page 95, 187.

which is exactly the same one given to student S3 as shown in Table 6.5. A correct response, if the student can still clearly remember rule [R609], might be as follows:

$$\log \frac{60}{6}$$

However, from what is shown in Table 6.5, we can see that student S3 did not respond this way. She seemed to be applying a rule (line 2) and this rule says "you can always change subtraction to division" which is even more general than that expressed by rule [MR609]. As the subject was applying a rule, there should be no impasse and thus no repairing. The error was not generated by an impasse-repairing process.

As the rule [MR609] is a variant of rule [R609], the two are different only in the action parts but are completely identical in their condition parts. If student S3 did learn rule [R609], then she would always give the correct answer. If she forgot the rule, there should be signs showing that repairing was ongoing. However, neither conditions seems to be the case reported here. Student S3 seemed to be applying a mal-rule [MR609] she learned when she was supposed to be learning the correct one [R609].

This leads back to VanLehn's (1990) suggestion that learning is never complete. However, just saying that learning is incomplete does not explain the learning process. Based on the analyses of the students' problem solving processes (Appendix R), a model for the explanation of incomplete learning is suggested below:

### **Frame System**

To better illustrate the model on incomplete learning, the concept of frames, which are commonly used in artificial intelligence to represent a collection of

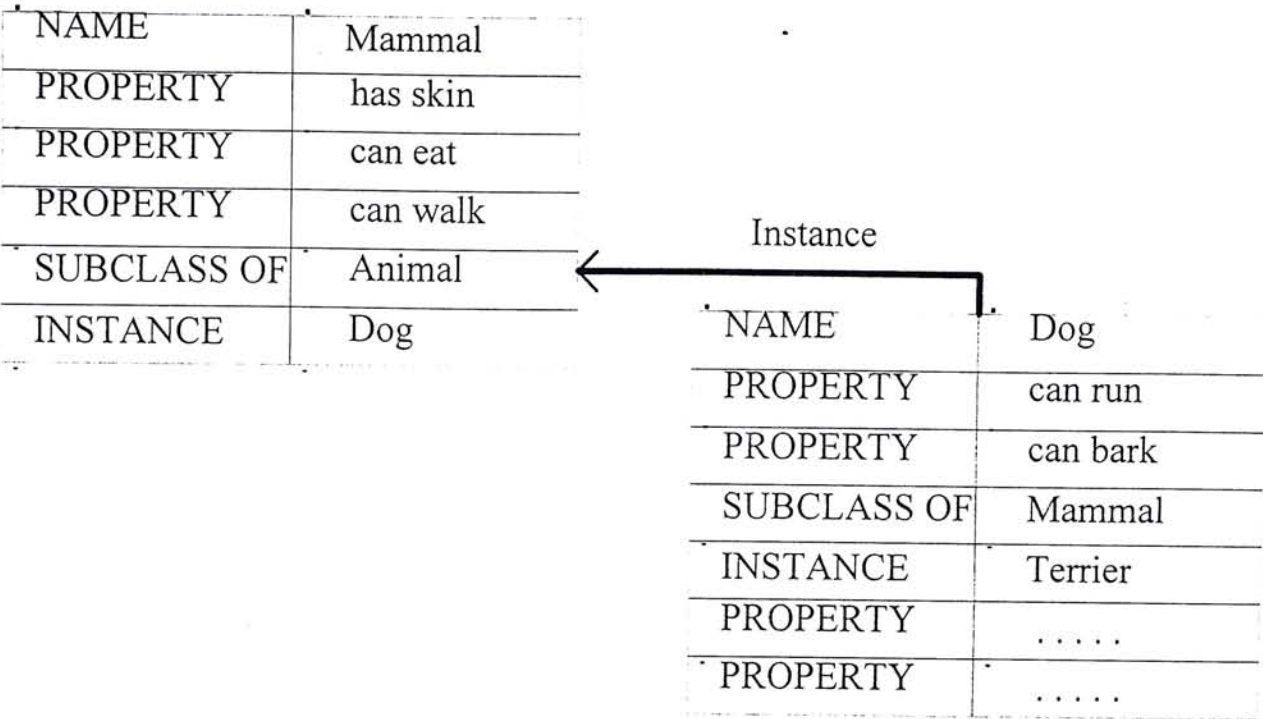


information, is introduced. According to Chabris (1991), a frame is a structure like a database record with slots and fillers corresponding to fields and values. The idea of frames was introduced by Minsky (1981) to represent a stereotyped situation, like being in a certain kind of living room, or going to a child's birthday part. Example below shows how the concept dog can be represented as a frame:

NAME	Dog
PROPERTY	can run
PROPERTY	can bark
SUBCLASS OF	mammal
INSTANCE	Terrier
PROPERTY	.....
PROPERTY	.....

In this frame, each row represents a field, called a slot. The first column shows the names of the slots and the second column shows the values. Hence the value of the NAME slot is Dog, which means that the name of the object represented by the frame is a dog. There may be several property slots used to describe the concept "dog". Some of the slots are filled. For example, can run and can bark. The others are to be filled when the frame is used or when it is joined with other frames which is described in the next paragraph.

Frames can be joined together as frame systems. For example, if we have another frame called mammal, then this frame and the previous one can be linked together as shown below:



The two frames in the above example are joined by a link called Instance which means a dog is an instance of a mammal. Two frames joined by an Instance link have a property that the instance frame (in this case, the dog frame) can inherit the properties of its class frame (in this case, the mammal frame). Thus, although the properties can eat and has skin are not listed in the dog frame, they are still properties of a dog as represented in this frame system. In these two cases, the two values are said to be the default values of the dog frame. A default value means that unless there is a contradictory value assigned to the same slot, the slot will take the default value. For example, if we have another frame for the concept whale, and since whales can only swim but not walk, then one of the slots for the whale frame should be filled with 'can swim' instead of the default value 'can walk' which is inherited from the mammal frame. For other frames that are instances of the frame mammal but do not have specific values for this property slot, 'can walk' will then be the default values.

There is evidence showing that we do, at least sometimes, use default values. An example given by Minsky (1981) serves to explain this. If we say "John kicked the ball", we would infer that this ball is of a particular size, particular color and weight

although these are not mentioned in the sentence. However, the default value of the color of the ball will be replaced by a particular value "white" if instead we say, "John kicked the white ball".

**Incomplete Learning**

Frames are effective means for the storing of knowledge. Productions used in representing the rules stated above can also be expressed in terms of frame systems. For example, the rule [R609] can be represented as follows:

NAME	Input Pattern		NAME	Output Pattern
NO. OF TERMS	2	→	NO. OF TERMS	1
OPERATOR 1	minus		OPERATOR 1	
TERM 1	log of Expression 1		TERM 1	log of Expression 1
TERM 2	log of Expression 2			over Expression 2
			TERM 2	

In learning this rule, it is necessary for the student to notice the differences between the input and the output patterns. These include:

1. Subtraction in the input pattern is changed to division in the output pattern but this time the division should be inside the logarithm;
2. There are two logarithms in the input pattern but only one logarithm in the output pattern;
3. The logarithms in the input pattern are the logarithms of single quantities (i.e.,  $A$  or  $B$ ) but the logarithm in the output pattern is the logarithm of a composite quantity (i.e.,  $A/B$ ), where  $A, B$  are any quantities.

However, when students are learning this rule, they would sometimes overlook some of these features. For example, a student might pay much attention to



the peculiar part of the rule stating that the subtraction in the input pattern has to be changed to division, but at the same time neglect all the other features. Thus, at that time, what goes into the student's working memory might be that represented by the frame system<sup>83</sup> as shown below::

NAME	Input Pattern		NAME	Output Pattern
NO. OF TERMS	2	→	NO. OF TERMS	
OPERATOR 1	minus		OPERATOR 1	division
TERM 1	log of Expression 1		TERM 1	
TERM 2	log of Expression 2		TERM 2	

The frame representing the output pattern contains several empty slots which should be the result of incomplete learning. The student only noticed that the minus should be changed to division when the input pattern contains logarithms, but did not notice what should be made on the number of terms and what the term should become. Thus, slots representing these features are now empty. There should be two possibilities when this system is stored into long-term memory. Either this frame system is stored with the empty slots or the system is stored with the empty slots filled with default values. In the first case, the learned rule would later become the source of impasse which then needs to be repaired<sup>84</sup>. In the second case, the most probable default values would be those inherited from the input pattern. The following shows what is stored in the long-term memory when this happens:

<sup>83</sup> A frame system is a way to represent knowledge in a systematic way. It does not necessarily imply that knowledge is stored in the working memory in the form of frames.

<sup>84</sup> When this rule is applied, because of the empty slots, the student would then get stuck (the impasse) and would have to find some means to fill up these slots in order to continue. The process of filling these slots can be thought of as some kind of repairing.

NAME	Input Pattern		NAME	Output Pattern
NO. OF TERMS	2		NO. OF TERMS	<u>2</u>
OPERATOR 1	minus	→	OPERATOR 1	division
TERM 1	log of Expression 1		TERM 1	<u>log of Expression 1</u>
TERM 2	log of Expression 2		TERM 2	<u>log of Expression 2</u>

\* Values of slots underlined are those inherited from the input pattern.

**Misperceiving when Learning**

Notice that the example above now shows exactly the error represented by rule [MR609]. Thus, in addition to those errors that are generated by misgeneralization<sup>85</sup> or by impasse-repairing<sup>86</sup>, the discussions above show a new type of error like [MR608], [MR609] that are generated by misperception. The errors are formed when students are perceiving (encoding information into working memory) the information to be learned. When the information is perceived incompletely, the missing values will be filled with default values that either come with the encoded information (as in the case above) or from the student's previous experience<sup>87</sup>. Errors are then formed when incorrect values are filled into the empty slots.

**Misperceiving when solving**

The discussion in Chapter 3 suggested that there are two types of errors: the off-line and on-line<sup>88</sup>. Since off-line errors are caused by using mal-rules that are generated when on-line errors were made, it is thus the on-line errors that have to be

<sup>85</sup> Refer to page 100 of chapter 3 for detailed discussion.

<sup>86</sup> Refer to page 96 of chapter 3 for detailed discussion.

<sup>87</sup> For example, the student might have learned that addition can be changed to multiplication (which is wrong). This would help him or her to fill in the values more easily.

<sup>88</sup> Page 104.



studied in order to investigate their cause. Although quite a lot of on-line errors can be explained by impasse-repairing processes<sup>89</sup>, there is still the possibility that there are other causes. It was argued in the previous section that errors may be caused when a rule is learned incompletely due to misperception. In the section below, it is further argued that errors may also be caused when the problem is being solved, again due to misperception. Misperception may be an important cause of errors besides impasse-repairing and misgeneralization.

### **Impasse-repairing and Rule-applying**

It should not be difficult to differentiate between the on-line and the off-line errors since the former are characterized by a problem solving process to repair an impasse while only rules are applied in the latter. As problem-solving process and rule-applying are processes that happen in the mind, it is not easy to differentiate between the two types of errors externally. However, as repairing an impasse is a problem-solving process that always takes longer than the process of applying a rule, a possible way to differentiate the errors is to observe the time required for the process. An incorrect response that is given very fast would most probably be an off-line error while an incorrect but slow response characterizes an on-line error. Furthermore, if a student is given a problem to solve and there is available an elementary rule with which every one is very familiar, and the student responses quickly with a wrong answer, and this error does not seem to be caused by careless work, then there should be no doubt that this error is an off-line error caused by rule-applying.

---

<sup>89</sup> Page 96 of chapter 3.



Such errors can be found from the students tested and interviewed in the present study. A typical example is that during the process of simplifying logarithmic expressions, a simple expression "0.4771 - 4.771" has to be simplified. As this should be very easy for secondary four students, they should have no difficulty in giving the answer "-4.2939". Surprisingly a student expressed it as  $\frac{0.4771}{4.771}$ . Similarly, another student was found to express  $\frac{2 \times 0.4771}{2 \times 0.301}$  as 0.9542 - 0.602. As  $2 \times 0.4771$  equals 0.9542 and  $2 \times 0.301$  equals 0.602, this student was actually making the same error as the first student. When expressed in the form of rules, these two errors become:

$$[E1] \quad \frac{0.4771}{4.771} \rightarrow 0.4771 - 4.771$$

$$[E2] \quad \boxed{\frac{2 \times 0.4771}{2 \times 0.301}} \rightarrow \frac{0.9542}{0.602} \rightarrow 0.9542 - 0.602$$

The rule [E2] stated above is a bit different from [E1] in that the first fraction enclosed in the dotted rectangle was the original expression, but it is believed to be equivalent to the one that follows. Hence the two rules are actually both instances of the rule:

$$[MR614] \quad \frac{Expression1}{Expression2} \rightarrow Expression1 - Expression2$$

There should always be rules, such as [R610] shown below, available for the students to evaluate the fractions,

$$[R610] \quad \frac{Number1}{Number2} \rightarrow Number3 \text{ (obtained by division)}$$

Yet, strange enough, the students did not use this rule which should be quite easy and convenient, but instead they used another rule in the form of [MR614]. To

investigate the reason for this kind of errors, the student who attempted the error represented by [E2] was interviewed<sup>90</sup>. Table 6.6 shows the part of the interview related to the error.

From line 6 in Table 6.6, it can be seen that the error was caused by the application of a rule although the student herself was not sure whether or not this rule was correct. However, it is interesting to note that there was no sign that she was using a rule in the form of [MR614]. Instead, the student seemed to think that since the expression to be simplified (i.e.,  $\frac{2 \times 0.4771}{2 \times 0.3010}$ ) was about logarithms, and she knew that when it is related to logarithms, division can be changed to subtraction (line 10 in Table 6.6), this was what she did. In other words, the student was actually applying a rule [MR615] shown below instead of the rule [MR614],

[MR615]

$$\frac{\text{Expression\_in\_log } 1}{\text{Expression\_in\_log } 2} \rightarrow \text{Expression\_in\_log } 1 - \text{Expression\_in\_log } 2$$

where *Expression\_in\_log1* and *Expression\_in\_log2* are two expressions involving logarithmic functions.

---

<sup>90</sup> For details, refer to Appendix R.

Table 6.6

Protocol of Student S3 that attempted error [E2]

---

1	Q:	You have $\log \frac{3^2}{2^2}$ here, how did you get the next two lines $\frac{2 \times 0.4771}{2 \times 0.3010}$ and $0.9543 - 0.602$ ?
2	A:	I took care of the indices first.
3	Q:	Do you mean that you did this first ( $\frac{\log 3^2}{\log 2^2}$ shown)?
4	A:	Yes. I then moved the indices to the front, and $\log 3$ equals this (pointed to $0.4771$ )..., so I could have $\frac{2 \times 0.4771}{2 \times 0.3010}$ . Then I didn't know how to find the ratio. Why did the division become subtraction? I just didn't know whether it is division or subtraction.
5	Q:	Division? What made you think that the division should become subtraction? It is interesting.
6	A:	I don't know. It seems that I was told that for the $\log$ , sometimes it can be division and sometimes it can be subtraction. So I.... Perhaps I was wrong, but I don't know why I did it that time.
7	Q:	So you now know that this is wrong, but at time, you just did it this way.
8	A:	Yes.
9	Q:	The reason is because there was something about $\log$ there?
10	A:	Yes. It is something like division becomes subtraction.

---



It should be quite clear that neither  $0.9542$  nor  $0.602$  involves any logarithmic functions and thus it is not appropriate to apply rule [MR615] to evaluate the ratio  $\frac{0.9542}{0.602}$ . However, from the example above, we can see that some students do treat any expression, even a simple real number, as expressions involving logarithms and then apply rules applicable to this perceived situation to solve the problem. Exactly why they do so needs further investigation. However, such errors have a characteristic that distinguishes them from the others: as there was no sign of impasse or repairing, this kind of errors should be some off-line errors caused by applying rules. However, the rule that caused the error was not a mal-rule. On the contrary, the rule that caused the error was a rule which was correct when applied to a suitable situation. In other words, it is the application of a correct rule to an incorrectly perceived situation that caused the error. In addition, the misperceiving happens when the problem is being solved and thus, should be differentiated from the errors caused by misperceiving when the related rules are being learned.

### **Misperception Versus Misgeneralization**

It should be noted that if we had not interviewed students to find out what they were thinking when the errors occurred, the same error could be explained in terms of misgeneralization. The rules [MR614] representing the error and the rule [MR611] are similar except that expressions in [MR611] are those involving logarithms. If the students further generalized the rule [MR611] by releasing the constraint that the expressions must be those involving logarithms, then the two rules would be identical.

In other words, the production P4 that corresponded to [MR611] can be generalized to the production P7 shown below which corresponded to the rule [MR614]:

P7    IF            *Expression1* minus *Expression2* is to be simplified,  
      THEN    express it as *Expression1* over *Expression2*

Again, an important point to note is that in this case, [MR614] has already become part of the knowledge of the student. This rule may be applied whenever it is required in future.

Hence, even for the same error, there can be two explanations: misgeneralization or misperception. Both misgeneralization and misperception are internal processes that cannot be easily detected. Further, even if we can detect that one student is misperceiving, we cannot say that all the other students are committing the same error. The reverse is also true if we can detect that a student is misgeneralizing. Hence, the best thing we can do for the time being is to accept both as causes of errors. However, there should be an important distinction between errors caused by misperception and misgeneralization: As misgeneralization results in the modification of rules that can be applied to more generalized situations (For example, the generalization from rule [MR611] to rule [MR614]), such modification of rules would cause the change on the condition parts. On the other hand, if the errors were committed through misperceiving, there is no need to change the condition parts of the knowledge rules since they still apply to the original situations though these situations might be perceived wrongly. The implication of such a distinction may have great effect on the system design but may be beyond the scope of the present study.

## **Meta-rules**

As for the system design, since the aim of the system is to help students to correct their errors, it is not necessary for the system to distinguish between the two causes of errors. Hence, it would be sufficient, for the time being, if the system is incorporated with only the meta-rules expressing the causes of errors, regardless of whether they are caused by misgeneralization or misperceiving. Table 6.7 shows the set of meta-rules that can be used to explain the mal-rules found in the present study. For the sake of simplicity, only mal-rules that happened not fewer than five times are listed and explained. As the prime interest for the present system is to diagnose and remedy errors committed by the majority of students, it might not be necessary for it to store rules to handle infrequent errors that arise from a variety of reasons. Those infrequent errors should be taken care of by the mal-rules discussed in the next paragraph..

The meta-rules are not designed to replace the mal-rules that can be generated by them. Instead, all the mal-rules that happened frequently were kept in the system in order to allow it to respond quickly to students' inputs<sup>91</sup>. The meta-rules are expected to handle the rare errors only.

---

<sup>91</sup>When there is a rule provided, the system can response by just applying the rule. If not, the response has to be generated by deducing or inferring from the meta-rules as well as the correct rules, which is thus much slower.



Table 6.7

List of mal-rules as explained by meta-rules

Code	Rule	freq- uenc y	Meta- rules
AA1	$\log(A \pm B) \rightarrow \log A \pm \log B$	64	MtR1
AA2	$\log A \pm \log B \rightarrow \log(A \pm B)$	23	MtR1
AA5	$\log Exp = -\log A \rightarrow Exp = -A$	14	MtR1
AA6	$\log Exp = 0 \rightarrow Exp = 0$	5	MtR1
AA7	$\log(A \times B) \rightarrow A \times \log B$	23	MtR1
AA8	$A \times \log B \rightarrow \log(A \times B)$	15	MtR1
AA10	$\log A \rightarrow A \times \log;$ $\log A \rightarrow \log \times A$	5	MtR1
AA12	$A \times \log B = \log C \rightarrow A \times B = C$	5	MtR1
AB2	$\log(A + B) \rightarrow \log A \times \log B$ or $\log(A - B) \rightarrow \frac{\log A}{\log B}$	19	MtR1 & MtR2
AB3	$\log(A \times B) \rightarrow \log(A + B)$ or $\log \frac{A}{B} \rightarrow \log(A - B)$	7	MtR3
AB4	$\log A \times \log B \rightarrow \log(A + B)$ or $\frac{\log A}{\log B} \rightarrow \log(A - B)$	7	MtR1 & MtR3
AB5	$\log A + \log B \rightarrow \log A \times \log B$ or	31	MtR2

	$\log A - \log B \rightarrow \frac{\log A}{\log B}$		
AB6	$\log A \times \log B \rightarrow \log A + \log B$ or $\frac{\log A}{\log B} \rightarrow \log A - \log B$	23	MtR3
AB7	$\text{Log. Exp.1} - \text{Log. Exp.2} \rightarrow \frac{\text{Log. Exp.1}}{\text{Log. Exp.2}}$	27	MtR2
AB8	$\frac{\text{Log. Exp.1}}{\text{Log. Exp.2}} \rightarrow \text{Log. Exp.1} - \text{Log. Exp.2}$	34	MtR3
AB9	$\text{Log. Exp.1} + \text{Log. Exp.2} \rightarrow \text{Log. Exp.1} \times \text{Log. Exp.2}$	10	MtR2
AB10	$\text{Log. Exp.1} \times \text{Log. Exp.2} \rightarrow \text{Log. Exp.1} + \text{Log. Exp.2}$	10	MtR3
AB11	$\log x \rightarrow x$ when not in an equation of the form $\log x = 0$	21	
AB15	$\log A \rightarrow \log(\text{value of } \log A)$	17	
AB16	$\log A^n \rightarrow (\log A)^n$	5	MtR1& MtR4
AB19	$\log \frac{A}{B} \rightarrow \log A + \log B$	8	
AB21	$\log A^n \rightarrow \frac{1}{n} \log A$	5	
AC1	$\log(A \times B) \rightarrow \log A \times \log B$ or $\log \frac{A}{B} \rightarrow \frac{\log A}{\log B}$	97	MtR1& MtR5
AC2	$\log(A + B) \rightarrow \log A + B$ or $\log(A - B) \rightarrow \log A - B$	17	MtR1& MtR5
AC4	$\log A + B \rightarrow \log(A + B)$ or	10	MtR1&

Knowledge Acquired

	$\log A - B \rightarrow \log(A - B)$		MtR5
AC5	$\log A \times \log B \rightarrow \log(A \times B)$ or	36	MtR1&
	$\frac{\log A}{\log B} \rightarrow \log \frac{A}{B}$		MtR5
BA1	unable to reject roots that cause log(-ve)	11	

---

Meta-Rules

Code	Meta-Rule
MtR1	Logarithms treated as multiplication with the variable "log"
MtR2	addition treated as multiplication or subtraction treated as division
MtR3	multiplication treated as addition or division treated as subtraction
MtR4	Incorrect index law used
MtR5	Incorrect distribute law used

---

Total Number of Mal-rules about logarithms	549
Number of Mal-rules explained by the Meta-rules	498
Percentage of Mal-rules explained by the Meta-rules	90.71%

---



From Table 6.7, as there were totally 549 mal-rules about logarithms collected and that 498 of them can be explained by just five meta-rules, it can be seen that most (90.71%) of the mal-rules can be explained by meta-rules. It should be pointed out that to generate the mal-rules, not only the meta-rules but also other correct rules are required. Further, besides the errors on logarithms made by the students, there were actually errors not concerned with logarithms and they were also found to be explainable by some meta-rules. As it is intended to focus on logarithmic errors in this study, there will be no detailed discussion on other types of errors. However, mal-rules concerning such errors were also collected and incorporated into Electronic Homework.

### Section Summary

This section describes results of the analysis of students' errors. While 549 errors were found in the students' tests, it was found that 90.71% of the errors concerning logarithms could be explained by using five meta-rules. Although these meta-rules could, in fact, replace most of the mal-rules, it was not done since by keeping the mal-rules, the system could react much faster to students' inputs.

Another important finding is that some of the errors in the present study were more appropriately explained by the students having misperceived the problem situation and thus used a rule, which was mostly correct in the perceived situation. This type of error was different from previous ones in that there seemed to be no sign of impasse repairing and the rules representing them could not be explained in terms of misgeneralization. Misperceiving as the cause of errors has never been reported and is therefore worth further exploration.

## Tutoring Module

In investigating the best tutoring strategy, effects of the three tutoring methods were compared. The effect of practice on students’ performance was also studied. Thus there were six categories of students being compared. A summary of the average duration of the six categories is shown in Table 6.8.

Table 6.8

Mean Duration (in minutes) by Tutoring Conditions and Practice Conditions

	Reteaching Only	MBR	CD
Without Practice	19.20	17.25	18.25
With Practice	25.57	23.60	19.63

Note. MBR = Model-based Remediation; CD = Conceptual Dissonance

Table 6.8 confirms the intuitive notion that students will need more time to do more practice. However, approximately the same time were used for students adopting any one of the three remediation methods,. This shows that if effects of remediation methods did yield results, these could only be attributed due to the methods used and not to the time used.

### Effects of tutoring

Table 6.9(a) shows the means and standard deviations of scores in the pretest, posttest and retention test by tutoring conditions.

Table 6.9 (A)

Test Results by Tutoring Conditions (N=50)

<i>Condition</i>	<i>Pretest</i>		<i>Posttest</i>		<i>Retention Tests</i>	
	<i>M</i>	<i>SD</i>	<i>M</i>	<i>SD</i>	<i>M</i>	<i>SD</i>
Reteaching	55.88	23.53	80.88	17.96	67.33	19.17
MBR	59.17	18.96	85.83	9.89	60.28	22.52
CD	59.41	15.60	89.41	8.99	73.24	16.00

Note. MBR=Model-based Remediation; CD=Conceptual Dissonance.

Maximum score = 100.

Raw data can be found in Appendix 5A



Table 6.9 (B)

T-scores between mean scores of different tutoring groups

	Pretest	Posttest	Retention Test
Reteaching Only (n=15)			
Pretest	--	4.93***	0.91
Posttest		--	3.55**
Retention Test			--
Model-Based Remediation (n=18)			
Pretest	--	5.85***	0.19
Posttest		--	4.66***
Retention Test			--
Conceptual Dissonance (n=17)			
Pretest	--	8.08***	5.00***
Posttest		--	4.46***
Retention Test			--
Overall (n=50)			
Pretest	--	10.69***	2.34*
Posttest		--	7.19***
Retention Test			--

Note. \*:  $p < .05$ , \*\*:  $p < .01$ , \*\*\*:  $p < .001$

Analysis of variance showed that there was a significant overall mean difference among the three tests for all groups ( $p < .01$ ). Post hoc analysis showed that mean scores in both posttest and retention test were significantly higher than those in the pretest ( $p < .01$  and  $p < .05$ ). The mean score in the posttest was significantly higher than that of the retention test ( $p < .01$ ). The difference between the posttest and pretest as well as the difference between the retention test and pretest clearly showed that the remediation methods were beneficial to students. Although the effects of remediation deteriorated in the retention test held three months after the posttest, the score was still higher than that of the pretest. This latter result is contradictory to that of Sleeman et al. (1993) who claimed that the overall mean scores for the delayed posttest (two months after the first posttest) had reverted to the pretest levels. In order to investigate this discrepancy, the three tests for individual tutoring methods were compared. The results showed that for all three tutoring methods, significant differences were found among the three tests ( $p < .01$ ). For the differences between pair of tests, the results of analysis are as shown in Table 6.9(b).

It can be seen that all the differences were significant except for the differences between the pretest and retention tests for the Reteaching and MBR groups. Hence, if according to Sleeman et al. only these two groups were considered, there should be no significant difference between the pretest and retention test. The significant difference between the pretest and retention test may be attributed to the CD method, which in a way, proves that CD is superior to the other methods in retaining the learning effect.

Scores in Posttest and Ceiling Effect

When the scores of the Posttest were taken for the different tutoring methods including the groups with and without practice, no significant difference was found among the six groups. As all the mean scores in the posttest were quite high, it is possible that these scores had reached the ceiling so that even better methods cannot show greater effects. Detailed discussion of this ceiling effect will be dealt with in later sections.

Effects of Practice and Tutoring Methods on  
Retention test

Table 6.10 shows the mean scores in the retention test by practice and tutoring conditions. Table 6.11 shows the results by using two-way analysis of variance with pretest as covariate.

Table 6.10

Mean Scores by different practice and tutoring conditions on retention test

	Reteaching Only	MBR	CD
Without Practice	63.50	48.89	71.11
With Practice	75.00	71.67	75.63

Note. MBR = Model-based Remediation group; CD = Conceptual Dissonance group.



Table 6.11

Analysis of scores in retention test by practice and tutoring methods with -pretest as covariate

Source of Variation	Sum of Squares	DF	Mean Square	F	Sig. of F
WITHIN+RESIDUAL	11543.49	43	268.45		
REGRESSION	3458.66	1	3458.66	12.88	.001
PRACTICE	1160.77	1	1160.77	4.32	.044
MTHDGRP	1434.93	2	717.46	2.67	.081
PRACTICE BY MTHDGRP	1126.22	2	563.11	2.10	.135
(Model)	7794.51	6	1299.09	4.84	.001
(Total)	19338.00	49	394.65		
R-Squared =	.403				
Adjusted R-Squared =	.320				

Significant difference was found between the practice and no-practice groups ( $p < .05$ ) which proves that practice does enhance tutoring effects. For the three tutoring conditions, no significant differences could be found. However, the significant level was not far from being acceptable ( $p = .081$ ). It had just been shown that practice improved the subjects' performance. However, if the subjects' performance had reached their ceilings, then tutoring with better methods, if any, would make no difference. Hence the ceiling effect might again attenuate the effect brought about by tutoring methods, especially for the rather low significant level of the present study. In order to clarify this, only the scores of those subjects who did not practise were analyzed. The result is presented in Table 6.12.

Significant differences among the three conditions were found ( $p < .05$ ). This clearly shows that when the effects of practice are removed, effects due to the tutoring methods will surface. This confirms the point that different strategies have different effects on tutoring and that the effects are attenuated by the effects of practice.

When different tutoring methods were compared, it was found that CD condition was significantly better than MBR ( $p < .01$ ). No significant difference could be found between any other pair of the three conditions.

Table 6.12

Analysis of Scores of Subjects without Practice in Retention test by Tutoring Conditions with Pretest as Covariate

Source of Variation	Sum of Squares	DF	Mean Square	F	Sig. of F
Covariates	2028.740	1	2028.740	6.491	.018
PRETEST	2028.740	1	2028.740	6.491	.018
Main Effects	2602.807	2	1301.403	4.164	.028
CONDITION	2602.807	2	1301.403	4.164	.028
Explained	4329.713	3	1443.238	4.617	.011
Residual	7501.537	24	312.564		
Total	11831.250	27	438.194		



## **Conclusion on What Strategy To Use**

The results of the analysis showed that students tutored under all the tutoring methods scored much higher in the posttest. Although the effect was found to have deteriorated between the posttest and the retention test, students still fared better in the retention test than they did in the pretest. This proves that all the teaching methods in general were effective in improving students' algebraic skills. Further analysis of the differences between pretest and retention test showed that significant difference could be found only in the CD group. This suggests that CD may be the best among the three strategies in helping subjects to retain the effects of learning for a longer time.

When different tutoring methods were compared, significant differences could only be found in the retention test but not in the posttest. This may be due to the ceiling effect. The ceiling effect (with the maximum score being 100) on the posttest scores is clear if we look at the overall mean scores which was found to be 85.39. For the six individual groups (three remediation methods either with or without practice), five of them obtained group means higher than 84. Considering that subjects' performances might be affected by factors such as attention span, slips and tiredness because the subjects had to take the test after school, these scores were actually at ceiling that any better tutoring method could not cause further improvement. Effects due to different methods were thus not observed.

On the other hand, when the retention test was taken after a long summer vacation, the average score decreased. At this time, the effects due to different tutoring methods and practice conditions could be observed. Although only different practice conditions showed significant differences at first, different tutoring methods

also showed their effects with further analysis when subjects with practice chances were ignored. It seems that besides affecting the posttest, the ceiling effect also affected the retention test. In this case, comparison among the different tutoring methods showed CD seemed to be the most effective method while MBR fared the worst.

In contrast to previous findings which showed no difference between MBR and Reteaching, this study found that the MBR was more effective provided the MBR should be equipped with Conceptual Dissonance. In addition to what ordinary MBR does, by just pointing out the errors based on the subject models inferred and then reteaching, students should be induced to understand how the causes of errors would contradict their previous knowledge and why the correct rules should be used. Further, immediately after the students had correctly solve the problems, they should be asked to practise similar problems to consolidate the knowledge just learned.

As Model-based Remediation and Reteaching are essential parts of ITS and CAI respectively, it seems that ITS should have better effects than its counterpart. It is thus suggested here that ITS would be a better tutoring tool than CAI if suitable conceptual knowledge can be incorporated.

## **Discussion**

The present study found that the effects of tutoring and practice conditions were attenuated by the ceiling effect. It seems that the ceiling effect occurred because the tasks used were quite easy for the subjects. To further investigate the situation, it is suggested that more difficult problems be used in later studies. Also, there are limitations on the generalization of the results obtained. As the test involved solving



algebraic equations with one variable tutored by human tutors, it is doubtful whether the results obtained are applicable to other types of problems and computer tutors. This needs to be clarified before the strategies suggested could be incorporated into an ITS.

There are two findings in this study which do not agree with that reported by Sleeman et al. (1983). The first one is that the scores in the retention test were found to be significantly better than those in the pretest while in the Sleeman et al. study, no significant difference was found. This discrepancy was further shown to have resulted from the effect of CD. Hence, the discrepancy may have resulted from the use of a method which the student had not come across before.

The second discrepancy comes from the fact that in the study of Sleeman et al., no significant difference could be found between MBR and Reteaching Only. Although the same result was found in the present study, it was discovered that MBR had the least effect on remediation when compared with the overall results. This is strange since the best strategy found, viz. the CD method, is also a model-based remediation. The only difference between the two is that in the CD condition, students were told why the rules used are incorrect and why the correct rules should be used. Hence, data obtained in the present study seems to suggest that just pointing out subjects' errors without telling them the reasons would have the adverse effect of blocking learning rather than helping it. Further investigation is required to reveal the reason behind this.



## How Experienced Teachers Perceive

Besides hypothesizing causes of errors and studying strategies to help correct the errors based on students' problem solving processes and experiments, it would be worthwhile to compare the results with the perceptions of experienced teachers. A convenient way to collect such knowledge would be by looking at what experienced teachers usually do when they come across such errors in their students' work - what they think the origins of the errors are and what are the best ways to help them. A questionnaire shown in Appendix G based on the mal-rules collected was designed for this purpose. The questionnaire consisted of three parts: teachers' personal information, factors affecting problem difficulties and estimates of problem difficulties. Five experienced teachers were asked to answer the questionnaire. Results collected were analyzed and reported in Table 6.13.

In Table 6.13, the reasons and remediation methods for each of the mal-rules were listed with their codes and the meaning can be found at the bottom of the table. Since not all the teachers suggested reasons or remediation methods for each rule, and some suggested more than one reason or methods, the number of reasons and number of remediation methods were thus not even for all the rules. Besides, some of the reasons suggested were too general to give any insight on the causes of error. For example, reasons such as "confusion about grouping terms" or "difficult problem" were rejected.

Table 6.13

Causes of errors and ways to remedy them as perceived by experienced teachers

Code	Rule	freq.	Reason <sup>a</sup>	Remediation Method <sup>b</sup>
AA1	$\log(A \pm B) \rightarrow \log A \pm \log B$	64	R1,R1,R1, R12,R1	M1,M1, M6, M2
AA2	$\log A \pm \log B \rightarrow \log(A \pm B)$	23	R1,R1,R1, R12,R1	M1,M1
AA3	$\log(A \times x) = B \rightarrow \log x = B / A$	0	R2,R1,R1,R1	M1,M1
AA4	$\log(A \times x) \rightarrow A \times \log x$	4	R2,R1,R1,R1	M1,M1,M2
AA5	$\log \text{Exp} = -\log A \rightarrow \text{Exp} = -A$	14	R2,R1,R1,R1, R1	M1,M1
AA6	$\log \text{Exp} = 0 \rightarrow \text{Exp} = 0$	5	R2,R1,R1,R1, R1	M1,M1
AA7	$\log(A \times B) \rightarrow A \times \log B$	23	R2,R1,R1,R1, R1	M1,M1
AA8	$A \times \log B \rightarrow \log(A \times B)$	15	R2,R1,R1,R1, R1	M1,M1,M2
AA10	$\log A \rightarrow A \times \log;$ $\log A \rightarrow \log \times A$	5	R1,R1,R1,R1, R1	M1,M1,M2
AA12	$A \times \log B = \log C \rightarrow A \times B = C$	5	R1,R1,R1,R1, R1	M1,M1,M2
AB2	$\log(A + B) \rightarrow \log A \times \log B$ or $\log(A - B) \rightarrow \frac{\log A}{\log B}$	19	R3,R8,R1, R12,R3,R1& R10,R3	M2,M3,M5,M2
AB3	$\log(A \times B) \rightarrow \log(A + B)$ or $\log \frac{A}{B} \rightarrow \log(A - B)$	7	R4,R8,R1 & R10,R4	M2,M3,M5,M5
AB4	$\log A \times \log B \rightarrow \log(A + B)$ or $\frac{\log A}{\log B} \rightarrow \log(A - B)$	7	R4,R8,R1& R10,R4	M2,M3,M5,M5
AB5	$\log A + \log B \rightarrow \log A \times \log B$ or $\log A - \log B \rightarrow \frac{\log A}{\log B}$	31	R3,R8,R1& R10,R3	M2,M3,M5,M5
AB6	$\log A \times \log B \rightarrow \log A + \log B$ or $\frac{\log A}{\log B} \rightarrow \log A - \log B$	23	R8,R1&R10, R4	M2,M3,M5,M5
AB7	$\text{Log. Exp.1} - \text{Log. Exp.2} \rightarrow \frac{\text{Log. Exp.}}{\text{Log. Exp.}}$	27	R8,R1&R10, R3	M2,M3,M5,M2

## Knowledge Acquired

AB8	$\frac{\text{Log. Exp.1}}{\text{Log. Exp.2}} \rightarrow \text{Log. Exp.1} - \text{Log. Exp.2}$	34	R8,R1&R10, R4	M2,M3,M5,M 2
AB9	$\text{Log. Exp.1} + \text{Log. Exp.2} \rightarrow \text{Log. Exp.1} \times \text{Log.}$	10	R3,R8,R1,R 3	M2,M3,M5
AB10	$\text{Log. Exp.1} \times \text{Log. Exp.2} \rightarrow \text{Log. Exp.1} + \text{Log.}$	10	R8,R1,R4	M2,M3,M5
AB11	$\log x \rightarrow x$ when not in an equation of the form $\log x = 0$	21	R1,R1	M5
AB15	$\log A \rightarrow \log(\text{value of } \log A)$	17	R5,R1	M3, M5
AB16	$\log A^n \rightarrow (\log A)^n$	5	R9,	M3, M6
AB19	$\log \frac{A}{B} \rightarrow \log A + \log B$	8		M3, M6
AB21	$\log A^n \rightarrow \frac{1}{n} \log A$	5	R9	M3, M5
AC1	$\log(A \times B) \rightarrow \log A \times \log B$ or $\log \frac{A}{B} \rightarrow \frac{\log A}{\log B}$	97	R2,R1&R10	M3, M5,M1
AC2	$\log(A + B) \rightarrow \log A + B$ or $\log(A - B) \rightarrow \log A - B$	17	R6,R11,R11, R1	M3, M5,M1
AC4	$\log A + B \rightarrow \log(A + B)$ or $\log A - B \rightarrow \log(A - B)$	10	R6	M3,M1,M1
AC5	$\log A \times \log B \rightarrow \log(A \times B)$ or $\frac{\log A}{\log B} \rightarrow \log \frac{A}{B}$	36	R2	M3, M1,M1
BA1	unable to reject roots that cause $\log(-\text{ve})$	11	R7,R7	M4, M6

<sup>a</sup>Meaning of Codes used in the Column for Reasons:

- R1:  $\log(A \pm B)$  treated as  $k \times (A \pm B)$
- R2: did not realized the meaning of  $\log A$  as a function and  $A$  can be composite expressions like  $2x$ .
- R3: Only memorize  $"+" \rightarrow " \times "$ ;  $"-" \rightarrow " \div "$
- R4: Only memorize  $" \times " "+"$ ;  $" \div " \rightarrow "-"$
- R5: Only memorize "A" as "value of  $\log A$ "
- R6: Laziness in writing brackets
- R7: Students did not know that  $\log(-\text{ve no.})$  is not defined.
- R8: mixed up or unclear about the concept of addition, subtraction, multiplication and division within or between "log".
- R9: not clear about the log of a quantity in index form



## Knowledge Acquired

R10:  $k \times (AB)$  changed to  $(k \times A) \times (k \times B)$  or  $k \times \frac{A}{B}$  changed to  $\frac{k \times A}{k \times B}$

R11: unaware of brackets

R12: closer to the perceived answer

<sup>b</sup>Meaning of Codes used in the Column for Remediation Methods:

M1: Remind student of the concept of log as a function.

M2: Give examples to show that the two sides are different;

M3: Practice

M4: Point out the error

M5: Remind the concept of “log” as a function, give examples to show that the two sides are different.

M6: Revise the correct rule

Although there were some discrepancies found among the teachers' suggestions on the reasons and remediation methods, the majority seem to be common to those suggested in Table 6.7 which shows the meta-rules used by the system to explain the mal-rules. For example, for the first subgroup of rules (AA1, AA2,...), the system used a meta-rule MtR1 to explain them, where MtR1 states:

[MtR1]        Logarithms treated as multiplication with the variable "log".

On the other hand, reasons given by the teachers were:

R1:      $\log(A \pm B)$  treated as  $k \times (A \pm B)$

R2:     did not realized the meaning of  $\log A$  as a function and  $A$  can be composite expressions like  $2x$ .

R13:    closer to the perceived answer

While both R1 and R2 are actually saying the same thing and are identical to MtR1, R13 suggests a different perspective on the problem. However, this suggestion was based on the example given to the mal-rule [AA1] which states:

$$\log 5 + \log 5 \rightarrow \log(5+5)$$

The teacher suggested this might have occurred because the student wanted to have  $\log 10$  in the next step which might then easily lead to the answer. Such a suggestion is reasonable, but it is true that the student is still treating the logarithmic function as the multiplication with a variable "log". Actually all the teachers were suggesting the same explanation to this group of rules as that given by the computer system.

By using similar arguments, reasons suggested by the teachers to explain the mal-rules can be categorized under the headings of the meta-rules used in Table 6.7.

Table 6.14 shows the result of such categorization:

Table 6.14 shows that the teachers' suggestions are largely compatible with the meta-rules used by the computer system. Besides rules, on the remediation methods suggested, it was found that the suggestions could be placed into four categorized as follows:

Restate Rule:	M1, M5, M6
Compare with Previous Knowledge:	M2, M5
Practice:	M3
Point out errors:	M4

All these methods are those included in the Conceptual Dissonance Method described above<sup>92</sup>. Thus we can say that these experienced teachers were actually using methods similar to those used in Electronic Homework to tutor their students. In a way, Electronic Homework is doing what experienced teachers are doing.

---

<sup>92</sup>Page 151



Table 6.14

Table showing Relation Between Meta-rules and Reasons Given by Teachers.

Code	Reasons Given by Teachers
MtR1	R1,R2
MtR2	R3,R9
MtR3	R4,R9
MtR4	R10
MtR5	R7,R11,R12

\*\* Teachers' reasons not included: R5, R6, R8,R13.

## Chapter Summary

Reported in this chapter is the knowledge acquired for the different modules through the processes described in the last chapter. For the expert module, a total of 555 strategic rules and 64 axiomatic rules were obtained and were incorporated into the computer system as the knowledge required to solve the problems given to students. For the student and tutoring modules, results obtained provided bases for the discussion on why students make errors and how to remedy them. The following two paragraphs summarize these findings and the suggestions that follow.

On the students' errors, 114 mal-rules were obtained. When we count only those happened frequently (not fewer than 5 times) and are on logarithms, there are 27 of them. Of the 125 students who participated the mal-rule collecting test, 549 of frequent errors were identified. Among these frequent errors, 498 (90.17%) of them were found to be explained by 5 meta-rules. Besides, these meta-rules were supported by the reasons given by five experienced mathematics to explain the errors described by the mal-rules. Thus, it seems that the use of such meta-rules would, in a way, replace a large portion of the mal-rules in the computer system. However, it would not be done since the keeping of the catalog of mal-rules would make the computer system work faster.

On the tutoring strategies, the result of the experiment showed that causing conceptual dissonance would be more effective than the other two methods Reteaching Only and MBR (Model-based Remediation which means to develop a model of the student based on the student's performance and then remedy). The views

of expert teachers opinions revealed that the above three methods were frequently used. Thus findings on tutoring strategies seem to support the notion that the method of trying to cause conceptual dissonance should be the method used in the Electronic Homework.

One last but may be quite important finding from the analysis of students' errors is that besides, the traditional explanation of errors, the impasse-repairing and misgeneralization, misperceiving may also be a cause of errors. Misperception here means that when the error is made, there is no impasse and repairing. The student just encodes the problem situation incorrectly and then uses a rule that should be applied in the perceived situation, to solve the present problem. In this case, the error is neither caused by impasse-repairing nor misgeneralization. Misperceived errors can be of two types: one is due to the incorrect perception of the rules when the rule is being learned and the other is due to the misperception of the problem situation. A model for the explanation of misperceived errors based on default values inherited from frames has also been suggested.

Misperception suggests a new way of explaining error generating. However, all the explanations given on causes of errors until now are based on internal processes which might be difficult to identify. Besides, if only correcting students' errors is required, it seems that an effective tutoring strategy such as Conceptual Dissonance used in the computer system would be sufficient for the purpose. The detailed study of misperceiving as causes of errors will be put aside as a topic for later studies.



## ***CHAPTER 7***

# **PROBLEM DIFFICULTY**

### **Overview**

In order to devise a measure of problem difficulty which can be used before a test is administered to the students, it was necessary to collect or calculate the item difficulty ratio, students' estimation of item difficulty level and teachers' estimation of item difficulty level. The item difficulty ratio is traditionally used to measure item difficulty but further investigation must be conducted to determine whether the other two measures can provide accurate predictions. Test results showed that the correlation coefficients between teachers' estimation and item difficulty ratio is 0.53 and that between students' estimation and item difficulty ratio is 0.86. Both can be considered as predictors of problem difficulty.

To investigate how teachers and students estimate problem difficulty, six cognitive difficulty factors were identified as follows:

1. Perceived number of difficult steps during the problem solving process
2. Number of steps required to finish the problem
3. Numerical complexity
4. Number of occurrences of "log"
5. Number of operations in the question

### 6. Students' degree of familiarity to the question

Before the teachers estimated the problem difficulty, they were requested to rate how these factors affected their estimations. The teachers were also invited to add any additional factors to the factor list. Data collected on these various measures were used to formulate a new measure of problem difficulty called *clevel*<sup>93</sup>. Correlation among the four measures: item difficulty ratio, teachers' estimation, students' estimation and *clevel*, showed the students' estimation, which was later accepted as the basis of the difficulty measure, generally had higher correlation coefficients with the others. A later multiple regression showed that the students' estimation could be predicted (multiple regression coefficient = 0.77) with some of the cognitive difficulty factors. A new variable called problem difficulty level developed on the regression equation would be used as prediction of problem difficulty in Electronic Homework.

## Results of Different Measures of Problem Difficulty

The results from analyses of the three different measures of problem difficulty, including the item difficulty ratio, students' estimation and teachers' estimation, showed that all the measures were highly correlated. Data obtained were analyzed and are reported in subsequent sections.

---

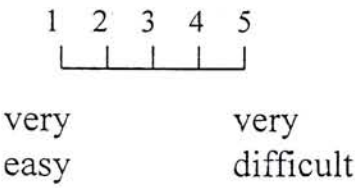
<sup>93</sup>Pronounced as "C level".

## Students' estimation of Item Difficulty

Students' estimations of item difficulty were collected together with the two mal-rule collecting tests (Appendix A). The item difficulty estimates were on a five point scale with one being the easiest and five the most difficult. For example, the first question in mal-rule collecting test 1 showed that:

1.  $\log 2 + \log 5$

and students were required to circle a number on the five-point scale. See example below:



where a value of 1 represents a very easy problem and a value of 5 represents a very difficult problem.

One hundred and twenty-five students responded to this part of the test and their estimates for each problem were averaged, the results are reported in column 3 of Table 7.1.



Table 7.1

Table showing various Mean Measures of Item Difficulty

Pa- per	Ques- tion	Students' Estimate	Item Difficulty Ratio	Teachers' Estimate	Predicted Complexity (clevel)
1	1	1.40	0.82	1.63	-17.05
	2	1.80	0.69	2.85	-8.56
	3	2.22	0.60	2.78	1.83
	4	1.90	0.78	2.11	-12.18
	5	1.71	0.74	1.81	-12.18
	6	1.85	0.62	2.26	-8.68
	7	2.28	0.52	2.56	-6.16
	8	1.62	0.82	2.74	-11.27
	9	1.84	0.60	3.11	-0.22
	10	2.46	0.49	2.85	-4.56
	11	2.12	0.62	2.44	-8.41
	12	2.66	0.38	2.78	1.91
	13	2.38	0.63	3.63	-2.88
	14	1.87	0.70	3.11	3.058
	15	2.77	0.28	3.22	22.77
	16	1.94	0.71	3.70	-13.52
	17	2.36	0.57	3.56	6.672
	18	3.17	0.10	3.74	21.54
	19	2.54	0.54	4.00	11.68
	20	2.53	0.40	3.26	-15.64
2	A1	2.28	0.43	3.33	-7.50
	A2	1.97	0.78	2.96	-9.90
	A3	1.86	0.73	2.56	-9.90
	A4	2.46	0.55	3.48	-1.62
	A5	2.33	0.70	3.67	15.18
	A6	2.56	0.59	3.89	24.02
	B1	2.87	0.34	3.44	-2.24
	B2	2.44	0.61	3.23	-0.87
	B3	2.45	0.64	3.11	0.50
	B4	2.45	0.33	3.37	7.12
	B5	3.17	0.37	4.00	9.59
	B6	3.36	0.27	4.33	27.45

**Item Difficulty Ratio**

Item difficulty levels were obtained from the result of the two mal-rule collecting tests. For each item, the item difficulty ratio was calculated with the following formula:

$$\text{Item difficulty ratio} = \frac{\text{Number of students with correct answers}}{\text{Total number of students}}$$

Results of the calculation are as shown in column 4 of Table 7.1 above.

**Teachers' Estimation of Problem Difficulty**

Teachers' estimations of problem difficulty were obtained through a questionnaire (Appendix G) consisting of problems from the mal-rule collecting tests. The questionnaire consisted of three parts: teachers' personal information, factors affecting problem difficulties and estimates of problem difficulties. Besides providing personal information and suggesting factors affecting problem difficulties, the teachers were also asked to rate each of the problems on the same five-point scale that students used<sup>94</sup>. Twenty-eight teachers whose work experience and qualifications are as shown in Table 7.2, were invited to participate in the test.

---

94 Page 245

Table 7.2

## Statistics of Teachers Contributed to the Estimation of Problem Difficulty

Teacher Characteristics		No. of teachers
Sex		
	Male	24
	Female	4
Age group		
	20-25	4
	26-30	14
	31-35	5
	36-40	2
	>40	3
Education level		
	Secondary	0
	Post-Secondary	0
	University Degree	15
	University Degree + Diploma of Education	7
	Master or Above	6
Mathematics as major subject studied		
	Yes	26
	No	2
Teaching experience:		
	0-2 years	3
	3-4 years	13
	5-6 years	5
	7-8 years	3
	more than 9 years	4
Teaching Experience (Secondary 3,4,5)		
	0-2 years	5
	3-4 years	12
	5-6 years	5
	7-8 years	3
	more than 9 years	3



Calculation from Table 7.2, showed that on average, the teachers who participated in the test had 5.03<sup>95</sup> years of teaching experience and among them, 4.70 years were in teaching secondary 3, 4 or 5 mathematics. Also, all of them were university graduates with 13 of them holding a Diploma of Education or a master degree. Hence, these teachers should be qualified to rate the different factors as well as problem difficulty. The estimations of these teachers on the difficulty level of each item were then averaged and reported in column 5 of Table 7.1.

### Predicted Complexity

In Table 7.1 a measure of the problem difficulty called predicted complexity was also shown. This measure was calculated on the six difficulty factors assumed in Chapter 5<sup>96</sup>. The six difficulty factors are: the perceived number of difficult steps, the number of steps required, the numerical complexity, the number of times 'log is used in the given expression, the total number of operations, and the degree of familiarity to the students. Before completing the questionnaire on estimating the problem difficulty (Appendix G), the same 28 teachers were requested to rate the relative importance of these six factors. They were encouraged to add other appropriate factors. The following paragraph describes the result of this investigation.

---

<sup>95</sup>In calculating teaching experience, 9 years or more were counted as 11 years for the sake of simplicity.

<sup>96</sup>Page 158.

## Teachers' Rating on Factor Importance in Predicting Problem Difficulty

To investigate how teachers estimate problem difficulty, six possible factors affecting the cognitive difficulty of problems were identified and presented to the teachers to rate their relative importance. Table 7.3 shows the averages of obtained from these ratings.

Table 7.3 shows the average levels of importance as perceived by the teachers for each of the factors denoted separately as  $f_1$ ,  $f_2$ ,  $f_3$ ,  $f_4$ ,  $f_5$  and  $f_6$ . Although these teachers were requested to add any new factor which they thought would affect their estimation, in fact, no new factor was suggested. On the importance of the original factors, all the levels of importance were found to be above average<sup>97</sup>, which shows that the difficulty factors were considered as quite important by the teachers.

To actually predict the problem difficulty by using these difficulty factors, it was necessary to collect the values of the factors for each problem. By using the procedures described in Chapter 5<sup>98</sup>, the data were collected and are reported in Table 7.4. As there were two different ways to calculate the number of steps required to finish a problem - that carried out by human experts (denoted by  $f_2$  (*human*)) and that carried out by the computer system (denoted by  $f_2$  (*computer*)) - they are both reported. The using of two different measures for the number of steps was to decide on which one should be used in the future by comparing their effectiveness in predicting the problem difficulty.

---

<sup>97</sup>The average is 3 since the highest is 5 and the lowest is 1.

Table 7.3

Teachers' rating on importance of factors affecting problem difficulty

Factor	Mean Level of importance
(f1) Perceived no. of difficult steps during the problem solving process	(r1) 4.00
(f2) No. of steps required to finish the problem	(r2) 3.43
(f3) Numerical complexity	(r3) 3.86
(f4) No. of occurrences of "log"	(r4) 2.96
(f5) No. of operations in the question	(r5) 3.21
(f6) Students Degree of familiarity with the question	(r6) 3.93



Table 7.4

## Problem Difficulty as Predicted by the Difficulty Factors

Paper No.	Q. No.	Factors								Predicted Complexity Level
		$f_1$	$f_2$ (human)	$f_2^*$ (computer)	$f_3$	$f_4$	$f_5$	$f_6$	(clevel)	
1	1	1	3	2.5	0	2	2	1	-17.05	
	2	2	3	3.0	1	2	2	1	-8.56	
	3	4	7	6.5	2	1	1	1	1.83	
	4	1	3	3.0	1	2	2	1	-12.18	
	5	1	4	3.0	1	2	2	1	-12.18	
	6	1	4	5.0	1	2	2	1	-8.68	
	7	3	6	4.0	2	1	1	1	-6.16	
	8	2	3	3.5	1	2	1	1	-11.27	
	9	2	3	1.5	2	3	3	1	-0.22	
	10	2	5	3.0	2	2	2	1	-4.56	
	11	3	6	5.0	1	1	1	1	-8.41	
	12	2	3	5.0	1	3	3	1	1.91	
	13	2	4	2.5	0	3	4	1	-2.88	
	14	3	3	3.0	3	2	2	1	3.05	
	15	3	6	8.0	4	3	3	1	22.77	
	16	1	2	2.0	2	2	1	1	-13.52	
	17	4	3	3.0	3	2	2	1	6.67	
	18	5	6	3.8	2	4	4	1	21.54	
	19	3	4	6.0	3	3	2	1	11.68	

# Problem Difficulty

	20	1	4	5.0	1	1	1	1	-15.64
2	A1	2	3	2.5	2	1	1	2	-7.50
	A2	1	3	1.5	1	2	2	2	-9.90
	A3	1	2	1.5	1	2	2	2	-9.90
	A4	1	4	2.3	1	3	3	2	-1.62
	A5	2	4	7.5	2	3	3	2	15.18
	A6	2	4	4.0	4	4	4	2	24.02
	B1	1	6	5.0	1	1	2	3	-2.24
	B2	1	4	3.5	2	1	2	3	-0.87
	B3	2	5	4.5	1	1	2	3	0.50
	B4	2	4	6.0	2	1	2	3	7.12
	B5	1	5	5.5	2	2	3	3	9.59
	B6	4	9	9.5	2	2	3	3	27.45

---

\*: Some values in this column are decimals since if only part of an expression is changed, it will be counted proportionally. For example, when “log 2 + 1” is expressed as “0.301 + 1”, it is counted as 0.5 step since only “log 2” but “1” is not.

As the desired measure for problem difficulty is to be calculated by the machine, the number of steps carried out by human experts is given for reference only. So long as the two measures agree, it is reasonable to use only the machine calculation. According to the above data, the correlation coefficient between the two variables was found to be 0.72 which should be high enough to ensure that there was the close relationship between the two variables. Hence all the remaining measures for problem difficulty were performed using on the machine measure only.

### Predicted Problem Complexity

As a test of whether the factors together can predict the problem complexity, a measure of human predicted problem complexity for the problems was calculated. The variable was referred to as *clevel* and was the weighted average of the values of the difficulty factors with teachers' rating on the relative importance (shown in Table 7.3) used as weights. A weighted average is different from a simple average which is obtained simply by dividing the sum of all values by the number of such values. For the weighted mean, there is a set of weights which indicates the relative importance of the values, and the weighted average is calculated by dividing the sum of products of values with the corresponding weights by the sum of weights. In other words, the formula for calculating the weighted mean is as follows:

$$\text{weighted mean} = \frac{\sum_{i=1}^n w_i x_i}{\sum_{i=1}^n w_i}$$



where  $x_i$  is the  $i^{\text{th}}$  value and  $w_i$  is its weights. For the calculation of *clevel*, as teachers' ratings correspond exactly to the relative importance of the difficulty factors, the same formula can thus be used as follows:

$$clevel = \frac{\sum_{i=1,3,4,5,6} f_i r_i + f_{2(\text{computer})} r_2}{\sum_{i=1,2,3,4,5,6} r_i}$$

with  $f_i$  representing the value of the  $i^{\text{th}}$  difficulty factor and  $r_i$  its corresponding importance as perceived by the teachers. Note that since there were two  $f_2$ 's but only that for the computer had to be included, hence in the numerator, there was no  $f_2 r_2$ ; instead a new term  $f_{2(\text{computer})} r_2$  was added.

## Correlation Among the Various Measures of Problem Difficulty

Before a detailed development of how the problem difficulty could be measured, it would be worthwhile to note the correlation among the various measures of problem difficulty mentioned above. Table 7.5 lists the correlation coefficients of the various measures of problem difficulty.

In Table 7.5, with the exception of the estimated problem complexity (*clevel*) which was calculated on the various difficulty factors, all the others were collected either by directly measuring students' performances (Item difficulty ratio) or through teachers' and students' self-reported estimation of problem difficulty (Teachers' and Students' estimation of Problem Difficulty). All the four measures were highly correlated.

Table 7.5

Correlation Coefficients Among the Measures of Problem Difficulty

	Dratio	Testm	Sestm	Ediff (clevel)
Dratio	1.00	-.53**	-.86***	-.61***
Testm	-.53**	1.00	.74***	.70**
Sestm	-.86***	.74***	1.00	.73***
Ediff (clevel)	-.61***	.53**	.72***	1.00

Note. Dratio = Difficulty Ratio; Testm = Teachers' Estimation; Sestm = Students' Estimation; Ediff = Predicated Problem Difficulty.

\*  $p < .05$ ; \*\*  $p < .01$ ; \*\*\*  $p < .001$ .

Among the various correlation coefficients, the highest one was that between item difficulty ratio and the students' estimation (-0.86). The negative sign in the coefficient represents the condition where the estimation is higher (more difficult), fewer students will answer the question correctly, hence causing a low item difficulty ratio. Comparing this coefficient with that between item difficulty ratio and teachers' estimation (-0.53) and that between estimated problem difficulty and item difficulty ratio (-0.61), it should be clear that the first one is highest. As item difficulty ratio has been traditionally used as the measure of problem difficulties, according to the above comparison, it seems that the students participants were able to better predict their own performances than the teachers. However, this result is not unreasonable since the estimation of problem difficulty by the students was carried out at the same time they were solving the problems. It would be natural for these students to rate as the difficult they could not solve and rate the others easier, which would then cause a high correlation between their ratings and the item difficulty ratio.

### How students rate the problems

Although it might be true that students would rate as difficult the problems that they could not solve and vice versa, it might also be true that they would rate such problems according to a finer degree of difficulty. For example, an unsolved problem could still be rated as quite easy if the failure was brought about by failure to remember a key formula. On the other hand, a solved problem could be rated as very difficult by a very bright student with very high self-esteem. In this way, the students' estimation of problem difficulty involves more than just seeing whether or not a



problem was solved. Rather, there might be some reflections on the problem's cognitive difficulty.

As the students' estimation on problem difficulty was found to be highly correlated with all the difficulty measures, it would be reasonable to say that this measure would be the most acceptable one among all the others. Hence, if this measure can be predicted by using some observable properties from the problem expressions such as difficulty factors described in the previous section, it would be possible to obtain a measure of problem difficulty before test administration. The following sections describe how this measure could be predicted and how the measure on problem difficulty was developed.

### **Predicting the Problem Difficulty Measures**

Table 7.5 showed the predicted problem difficulty<sup>99</sup> as well as all the other three measures collected directly from teachers or students. This information was not available when Electronic Homework was put into real use. Hence they have to be predicted by statistical methods. A statistical method called multiple linear regression was employed to investigate the effects of predicting the four difficulty measures by using the difficulty factors.

Multiple linear regression is a statistical method that tries to develop an equation that relates the given dependent variable to the various independent variables. If the dependent variable is  $y$  and the independent variables are (say)

---

<sup>99</sup> Page 242.

## Problem Difficulty

$x_1, x_2, x_3$ , then multiple linear regression can find out the constants, called regression coefficients ( $r_1, r_2$ , and  $r_3$ ) that can link up the variables in the form of:

$$y = r_1x_1 + r_2x_2 + r_3x_3 + C$$

where  $C$  is a constant in the above equation. Hence, by using this equation, the value of  $y$  can be predicted whenever the values of the independent variables  $x_1, x_2, x_3$  are given. Besides this equation, called the regression equation, there is also a constant called multiple regression coefficient which ranges from zero to one. The closer this value is to one, the more accurate this equation predicts the true values. For example, Table 7.6 shows that the regression coefficients of Item Difficulty Ratio and Students' Estimate, when predicted by the 6 difficulty factors with number of steps required by the machine to complete the problem as the basis of the calculation of the number of steps required, are 0.52 and 0.81 respectively. This reveals that Students' Estimate can be better predicted by the 6 difficulty factors than the Item Difficulty Ratio.

Hence, from the multiple regression coefficients shown in Table 7.6, we can see how the four measures of problem difficulty can be predicated by the different combinations of difficulty factors.

Table 7.6

Summary of multiple regression coefficients found

Number of difficulty factors	Problem difficulty measures			
	Item Difficulty Ratio	Students' Estimate	Teachers' Estimate	Predicted Problem Difficulty
6 factors (machine)	0.52***	0.81***	0.72***	1.00***
6 factors (human)	0.58***	0.80***	0.72***	0.98***
5 factors (machine)	0.52***	0.77***	0.72***	0.97***
5 factors (human)	0.58***	0.80***	0.72***	0.96***

\*p<.05. \*\*p<.01. \*\*\*p<.001.



Table 7.6 shows the multiple regression coefficients found in the analyses with the four problem difficulty measures as dependent variables and the six difficulty factors<sup>100</sup> as independent variables. Since there were two different sets of data regarding the number of steps required to finish the problem- one carried out by the machine and the other by human experts - there are two separate sets of results showing 6 factors denoting the machine and 6 factors denoting the human experts. Also, as it was found that the computer took too long to calculate the perceived number of difficult steps, it was anticipated that to realize a fast response to students' input, this factor might have to be excluded in the calculation. Calculation of coefficients shown in the last two rows of the table was to show whether excluding this factor would affect the prediction.

All the six difficulty factors assumed in the present study were found to be able to predict all the four measures of problem difficulty ( $p < .001$ ) though to different degrees of accuracy. Also, although the predicted problem difficulty was found predictable to a very high degree (multiple regression coefficient = 1), it is not a valid prediction since it was calculated by using the same difficulty factors that were used to predict it. For the other three measures, it would be interesting to look at the differences among them.

### **About the Three Measures**

For the other three difficulty measures, the item difficulty ratio, students' estimation and teachers' estimation, it was found that when they were predicted by

---

<sup>100</sup>Refer to page 238 of this chapter and also page 158 of chapter 5.

multiple linear regression with the six difficulty factors<sup>101</sup> as independent variables, not all of them had to be included in the regression equation. This shows that some of the factors might not be usable in the prediction of problem difficulties. Table 7.7 shows the difficulty factors that appeared in the regression equations to predict the various difficulty measures.

While the item difficulty ratio could be predicted with just one variable viz. the number of steps, when the students' estimation was predicted, three additional variables were required viz. the degree of familiarity, the number of operations and the number of perceived errors. This proves the assertion made earlier<sup>102</sup> that when students are rating the problem difficulty, they would do more than just rate as difficult those they could not solve and those they solved as easy. The calculation of item difficulty ratio was based on the number of students who could complete the problem. Hence, the more steps required in a problem, the more chances there were of making errors. That may be the reason why the number of steps alone can predict the item difficulty ratio. On the other hand, the fact that students' rating of problem difficulty depends on three additional variables suggests that they based their ratings on firstly, whether the problem was familiar to them, secondly, how complex the problem looked, and finally whether there were some perceived difficulties in which they would easily make errors. The students' estimations were found to be related more to the cognitive structure of a problem than the item difficulty ratio.

---

<sup>101</sup> Refer to page 242 of this chapter and also page 158 of Chapter 5.

<sup>102</sup> Page 245

Table 7.7

Variables in the Equations to Predict the Problem Difficulty Measures

Item	Difficulty	Students' Estimation	Teachers' Estimation	Predicted Problem
Ratio				Difficulty
MACHSTEP		FAMILAR	FAMILAR	MACHSTEP
(or HUMANSTEP)		MACHSTEP (or	NOLOG	(or HUMANSTEP)
		HUMANSTEP)	NUMCOMP	FAMILAR
		NOTMFAC		NOTMFAC
		PERERR		PERERR
				NUMCOMP
				NOLOG

Note. spercep = Students' estimation of problem difficulty; machstep = Number of steps required for the computer to finish the problem; humanstep = Number of steps required for human expert to finish the problem; familiar = Familiarity of the problems to the students; notmfac = Number of operator in the problem expression; pererr = Perceived no. of difficult steps during the problem solving process; numcomp = Numerical complexity; nolog = No. of occurrence of "log".



It has already been pointed out that arranging problems in terms of cognitive difficulty would be more helpful to students. It is thus possible that the student estimation of problem difficulty might be the better measure when compared with the item difficulty ratio which only represents the number of correct responses.

Another interesting point came from the factors predicting the teachers' estimation which was found to be dependent on only three factors: degree of familiarity, the number of 'log' and the numerical complexity, all of which are easily observable just from the problem expressions. Those factors that required in-depth study of the problems, such as the number of steps required to solve the problem, were not found in the equation. This revealed one important thing viz. when the teachers estimated the problem difficulty, they based their judgment on some easily obtainable and superficial variables. That might be the reason why their prediction of the student's achievement was not as good as those predicted by the students themselves. Be that as it may, we cannot deduce that this is what teachers usually do when predicting problem difficulties. But as far as the present study is concerned, the teachers' estimation should not be a good measure of problem difficulty.

### **Practical Considerations**

Analyses of the results above showed that among the four measures of problem difficulty, students' estimation would be the best. Our final task is to estimate this measure with the factors available. Four factors - the degree of familiarity, the number of steps required, the number of operations in the problem expression and the number of perceived errors - were found in the equation to predict the students' estimation. The easiest way to find the difficulty measure would be by using the

regression coefficients found in the regression equation in predicting the students' estimation. However, before this could be done, there are two practical considerations.

### **Machine Steps or Human Steps?**

Separate regressions were done for the number of steps required by computers or by human experts and the regression coefficients in predicting students' estimation were found approximately equal (0.81 and 0.80 for machines and human experts respectively) and highly correlated. Hence, it should be reasonable to say that using either one would yield identical results. As Electronic Homework is designed to arrange the problems in terms of difficulty immediately after the problems are entered, there would not be any time for any human expert to solve the problems before they are presented. The machine should count the number of steps and this should be used in the regression equation.

### **Should Perceived Errors be Included?**

Another consideration is the practical use of the computer system. The number of perceived errors is to be obtained by counting the possible chances of making errors. When this was attempted, it was found that twice the amount of the original time was required for making responses. As the present system has already been found to be rather slow in handling some difficult cases, adding this routine to count the number of perceived errors would increase the workload of the computer. Further, according to Table 7.6, it was found that when the factor number of perceived errors was removed, the multiple regression coefficient was 0.77, make it only slightly lower than the 0.81 obtained before. It would then be reasonable to remove this troublesome factor from the equation without sacrificing the accuracy of the prediction.

## Problem Complexity

A new variable was thus developed to represent the problem difficulty based on the students' estimation. It has been revealed that students' estimation could be predicted by several difficulty factors which roughly correspond to the cognitive difficulty of the problems. This predicted students' estimation could thus be a measure of the cognitive difficulty of a problem. As this predicted value should be different from the original students' estimation, and this measure was found to reflect the complexity of a problem, it was thus named differently as problem complexity. The finding of this problem difficulty depended on how the students' estimation could be predicted. The following shows the result of an analysis using multiple regression with students' estimation as dependent variable and the five difficulty factors, excluding the number of perceived errors, as the independent variables. The problem complexity was then defined according to the regression equation obtained.



Table 7.8

Summary of Multiple Regression Analysis for Variables Predicting Students' Prediction of Problem Difficulty (N=125)

Variable	r#
machstep	.11***
notmfac	.19**
familiar	.17*
(Constant)	1.17

Note:-  $R^2 = .59$ ,  $R = .77$ .

spercep = Students' Estimation of Problem Difficulty; machstep = No. of Steps required for the computer to finish the problem; familiar = Students' Familiarity with the Problems; notmfac = No. of operators in the problem expression; numcomp = Numerical Complexity; nolog = No. of Occurrence of "log".

# Correct to 2 decimal places.

\* $p < .05$ . \*\* $p < .01$ . \*\*\* $p < .001$ .

Table 7.9

Correlation Among the Problem Difficulty Measures (N=32)

	dratio	testm	sestm	ediff	compx
dratio	--	-.53**	-.86***	-.61***	-.58***
testm		--	.74***	.70**	.58***
sestm			--	.73***	-.77***
ediff				--	.82***
compx					--

Note. Dratio = Item Difficulty Ratio; Testm = Teachers' Estimation of Problem Difficulty; Sestm = Students' Estimation of Problem Difficulty; Ediff = Estimated Problem Difficulty; compx = Problem Complexity.

\*p<.05. \*\*p<.01. \*\*\*p<.001.

Table 7.8 above shows that the students' estimation could be reasonably predicted ( $R=.76596$ ) with the three difficulty factors: the number of steps, the number of operations and the degree of familiarity of the problems to the students. Hence, the problem complexity, which is the predicted value of the students' estimation of problem difficulty, was constructed according to the regression coefficient shown in Table 7.8. The equation of problem complexity was then developed as follows:

$$\text{problem complexity} = 0.11 \times \text{machstep} + 0.19 \times \text{notmfac} + .17 \times \text{familiar} + 1.17$$

Correlation coefficients between this new measure with each of the previous measures may be found in Table 7.9.

Table 7.9 shows that this problem complexity correlates significantly with the other difficulty measures and should, to a great extent, reflect the problem difficulty. Further, if this problem complexity is to be compared with the teachers' estimation, it can be seen that the correlation of this problem complexity with the other measures was higher than that of the teachers' estimation to the others, which then implies that the problem complexity could predict even more accurately than the teachers', at least for the problems in the present study.

## Using Problem Complexity in Electronic Homework

Problem complexity was used to calculate the difficulty of each problem entered into Electronic Homework. The calculated value was stored with the problem



and would be used as basis for arranging the problems in increasing order of difficulty in a problem set.

## Chapter Summary

Although problem difficulty can be measured by different means, data obtained in the present study showed students' estimation would be the best one among the four investigated. Students' estimation was superior to the others due to the fact that, in one way, it was found highly correlated with the others, and in the other, its measure included factors that represented the cognitive structure of the problems. However, as this estimation was calculated when the students were solving the problems, it would be impossible for Electronic Homework to obtain such measures before the problems were presented to students. A multiple regression was then carried out to predict the students' estimation by using three factors: the number of steps required, the number of operations in the problem expression and the students' degree of familiarity with the problems. Based on the regression coefficients thus obtained, a new variable called the problem complexity was defined and would be used as a measure of problem difficulty for the problems in Electronic Homework.

## ***CHAPTER 8***

# **SYSTEM EVALUATION**

## **OVERVIEW**

Whenever theory is put to practice, there are always difficulties that impede the application of a system to real situation. This chapter discusses the difficulties encountered during the developing of Electronic Homework and the subsequent application of the system. In the tests conducted, Electronic Homework was able to explain 90.71% of students' errors<sup>103</sup>, but when applied to real situations, the difficulty mainly came from the remaining unexplained ones. A number of factors affected the system, rendering it no better than the traditional homework method in promoting learning. Among these are the availability of sophisticated hardware in Hong Kong schools, the inadequacy of the software language that is used to build up the interface and even the time constraints on the testing of the system. However, even under these unfavourable circumstances, it was found that Electronic Homework did help students of one participating school to achieve better results in less abstract problems. Why this finding occurred only in one school was not explained by the data collected in the present study. One possible explanation may be found in the students' motivation

---

<sup>103</sup>Page 206 of Chapter 6.

towards learning. This chapter ends by discussing the results of the evaluation and the various ways to improve the system.

# The Evaluation

In order to check whether Electronic Homework is helpful to students, the effect of using the system has to be evaluated. The evaluation was done in two stages: the formative evaluation and summative evaluation. The formative evaluation was carried out at the development stage to debug (to find out errors) the system and to find ways to improve it, the summative evaluation was done after the system was finished with the purpose of investigating its effect on students' work. The exact procedures in doing these may be found in Chapter 5. The following sections describe the results obtained from these two evaluations.

## Formative Evaluation

The formative evaluation was carried out continuously over the 4 years taken to develop the system. To test whether the system can independently solve problems in logarithms, different problems from text books were entered. Any problem it could handle by itself would be taken care of by the researcher who either added new rules into the system or modified existing rules. To test the diagnosing ability of the system, students' errors, especially those obtained from the mal-rule collecting tests<sup>104</sup>, were entered into the system. The purpose was to see whether the system could correctly

---

<sup>104</sup>Test administered to 125 students to collect their errors. Refer to page 144 for more details.



identify them. If an error was not identified by the system, new mal-rules would be added to the system or some existing rules would be modified.

The system was also tested among four Hong Kong Secondary 3 students of average mathematical ability. Their processes of using Electronic Homework were video-taped for further analysis. The students were also required to complete a questionnaire on their perceptions on using Electronic Homework. Results of the analysis are discussed in the following section.

### **Students' Perception**

A summary of these students' perception on using Electronic Homework can be found in Appendix K. The results showed that these students found the system helpful in two ways: firstly, the feedback from the computer helped them to better master the learned materials, and, secondly they could learn at their own pace. They indicated that they would recommend the system to other students. Although this was only a very small sample which might not yield useful implications, this result does indicate that at least some students found it helpful.

### **System Testing**

In all the situations that the system was tested, besides finding that some new rules had to be added and some rules had to be modified, a common phenomenon was that the system worked too slowly, especially when an unfamiliar expression was entered. These unfamiliar expressions might correspond to infrequent errors (errors that occurred fewer than 5 times in the mal-rule collecting test were considered as rare

and were not represented explicitly in the system<sup>105</sup>) which then required the system to infer its origin and then generate the diagnosing message in real time. As inference takes time, the system thus reacted rather slowly. However, in most cases, these unfamiliar expressions were those that could not be explained by the rules already stored in the system. Hence the system has to take time to try out every means before it could admit that the expression entered was unexplainable. It was these undiagnosed errors that caused the slow response of the system at most of the time.

### Undiagnosed Errors

Electronic Homework stores all the correct rules required to solve problems in logarithms and the mal-rules required to diagnose the frequent errors (errors that occurred more than 5 times in the mal-rule collecting test). Every expression entered that can match one of these rules will receive a quick response since it does not take much time to search once all the rules are in the system. It is when a corresponding rule (correct or incorrect) cannot be found in Electronic Homework that there is a “slow response”.

There are 555 strategic and axiomatic rules in Electronic Homework and a “not diagnosed” message will be returned only after all the rules are searched and no match can be found. Therefore, the response given to an undiagnosed expression is usually slower than that given to a “diagnosed” error or that given to a correct expression.

---

<sup>105</sup>Refer to page 145 for more details.

The reaction of the computer will be even slower when the expression entered requires the system to use more than one of its rules to generate the response. For example, if the given problem is “log 6” and a student enters “log 2+ log 3”. There is no such rule in the rule base of the system saying that expressing “log 6” to “log 2+ log 3” is right or wrong. The computer then has to find a rule that can evaluate “log 6” as a new expression such as “log  $2 \times 3$ ” and a rule that can evaluate this new expression as “log 2+ log 3”. If two such rules are found and both of them are correct rules, the computer gives a “correct” message. Moreover, if one or both of these rules are incorrect, the computer gives an “incorrect” message.

This kind of problem that the computer has to use two or more rules to generate the expressions entered by students is referred to as “jump step” problems. These “jump step” problems stimulate the need for a discussion on whether the system should be designed so that it can store as much knowledge as possible or it should be designed to react as fast as possible. The following sections describe this in more detail.

### **Less memory space with composition of rules**

The method of combining two rules to explain an expression as it is done in the above section is referred to as composition by Anderson (1990, 1993). The composition of rules enables the computer system to store fewer rules without any loss of its diagnosing power. The consequence will be that the computer memory space can be saved for incorporating new knowledge, which then enables the system to be a tutor in a wider subject area.



### **Longer time required**

It takes shorter time for the computer system to respond to correct expressions or frequent errors. On the other hand, it takes much longer to identify an “undiagnosed expression. If the computer system is designed so that it does not take care of “jump step” problems, only 555 rules have to be scanned before the system admit that the expression entered is undiagnosed. If the system is designed to take care of “jump step” problems, then even for a “two-step” problem (two rules are to be composed), a total of  $555 \times 555$  compositions has to be checked, not to mention those with more steps. Hence, although composition of rules is a way to save memory space, it is done at the expense of response time.

### **Increase of speed at the expense of memory**

Originally, the design of Electronic Homework allowed the composition of two rules (“2-steps”). This design is not at all sophisticated since expert solvers sometimes jump three or four steps<sup>106</sup>. However, even with this limited capability, the system was found to be slow in some cases. Impatient students found the slow response of the system unacceptable. It was thus necessary for the system to increase its speed even at the expense of other factors like computer memory space.

The easiest way to solve this problem is by incapacitating the composing ability of the system. By disabling the composition component of the system, it can react much faster since only 555 rules to be searched every time an expression was

---

<sup>106</sup>For example, when asking to simplify the expression  $\log(9/4)$ , a student put down  $2(\log 3 - \log 2)$ , which if done step by step, should be in a sequence of  $\log(3^2/2^2)$ ,  $\log((3/2)^2)$ ,  $2\log(3/2)$  and  $2(\log 3 - \log 2)$ .

entered. But if this is done, all expression that require two or more rules to verify will become undiagnosed. The system loses a lot of its diagnosing ability.

This difficulty was overcome by composing rules before students' expressions are to be checked. For example, initially, the system stored two rules:

$$[R81] \quad \log X = \log Y \times Z \quad ; \text{where } X = Y \times Z$$

$$[R82] \quad \log X \times Y = \log X + \log Y$$

A new rule composed of rules [R81] and [R82] was then generated as follows:

$$[R83] \quad \log X = \log Y + \log Z \quad ; \text{where } X = Y \times Z$$

By incorporating these kinds of composite rules, the expressions entered corresponded to these composite rules received faster response than before since there is no need to compose rules during the interaction. However, the adding of composite rules used up memory space. Hence, the increase of speed was achieved at the expense of memory space. Besides, the diagnosing ability of the system would depend on the number of composite rules incorporated; the more composite rules were added, the higher the diagnosing ability, but the more inroad would be made into the memory space left for other knowledge pieces.

It should also be noted that the composing of rules in the present system is actually the same as skill acquisition suggested by Anderson (1990, 1993). In Anderson's theory, skill acquisition is the process of acquiring procedural knowledge and acquired skill is characterized by its speed and automaticity. In the present system, composing is also a way of increasing the processing speed of the system. The system with more composite rules reacts faster than those with fewer such rules. The development of this system can be thought of as a metaphor of human skill acquisition

process. An expert is a system with many composite rules while a novice is a system with only basic rules. With more and more composite rules added, a novice will become an expert.

### **The Windows Interface**

With the help of the computer language Visual Basic, the original text-based program written in Prolog could be equipped with a Windows interface. This Windows interface provides a more user-friendly environment for the users and also allows the users to enter their expressions by using a mouse pointer. Figure 8.1 shows the interface that students used to enter their expressions.



Problem		log	0					
current_problem		+	1					
		-	2					
		X	3					
		/	4					
		平方	5					
		立方	6					
		開方	7					
		指數	8					
		( )	9					
		(	.					
		)	x					
		or	y					
		=	刪除					
		此行完成	此題完成	列出公式,常數	提示	上一題	下一題	離開
		請逐步輸入,否則電腦可能不能判斷你是對或錯.						

Figure 8.1 Student-computer Interface in Electronic Homework

The use of a mouse pointer enables students who do not know how to type to enter their expressions easily. By just pointing the mouse to the symbol required and clicking the left button of the mouse, the symbol will be automatically entered. For Secondary 3 students, this may be more convenient than typing in the expression. However, when this input method is compared with the traditional pen-and-paper method which has been used for so many years, the latter might still be the more favoured one. If this is the case, the effect of using Electronic Homework will be reduced. This is why an item was included in the questionnaire on students' perception to test whether students would find difficulty in using the mouse or keyboard as input media.

A further drawback of this mouse-clicking method is that although students do not have to type in the expression, it is still not possible for them to enter expressions in exactly the same way as they appear in text-books. There are two reasons for not incorporating this technique into the system. Firstly, everyone who has experience in using systems such as the equation editor of Microsoft Word would have noticed that it takes quite a long time to load the system before you can actually type in the equation. In addition, it also takes a long time to unload it to go back to the original word processor, not to mention the tremendous amount of memory space in the computer that would be used up. Such time and memory requirements are not feasible for Electronic Homework. The second, and maybe the most important reason is that the developing of such an equation editor is itself a major project. Even large companies like Microsoft would require a long time to complete it. It is thus not easy to develop even a comparable one within this short period of time in the present study.

Although it was anticipated that the input method may not be well-received by the students, it was still put into real use, the reason being that Electronic Homework is not a finished product and the purpose of the evaluation is to find out which aspects of the system can be improved. Besides, the evaluation of the input method, can in fact, be part of the study. However, a more advanced input methods should be investigated in due course.

### **Language Interface Problem**

Electronic Homework may be the first computer program used under Chinese Windows that links the text-based computer language Prolog and the Windows-based computer language Visual Basic. There have been a lot of technical problems that required tremendous effort to solve. Most of the technical problems are now solved. However, a most notable problem discovered during the evaluation process that has yet to be solved is that the Visual Basic system seems to be quite unstable. Occasionally, a “General-protection Error” would suddenly appear and that when this happens, everything has to be started again. Electronic Homework is so designed that all expressions entered by the student will be stored and the student would, at most, lose only a few lines of the work entered. However, when this happened several times, students were frustrated, thus reducing the effect of Electronic Homework.

Despite so many factors unfavourably affecting Electronic Homework, including the speed problem, the Windows interface and the input method, the summative evaluation was still carried out in the hope that at least some students may find the system useful.



## **Summative Evaluation**

After the system was developed, it was trialled in six schools. Six classes of Secondary 3 students<sup>107</sup> in Hong Kong participated in the experiment. A total of 220 students were required to attempt two homework sessions: Homework 1 and Homework 2 (Appendix O). Ninety-two of the students took Homework 1 with Electronic Homework and Homework 2 with pencil and paper. Another 98 students took Homework 1 with pencil and paper and Homework 2 with Electronic Homework. Electronic Homework requires at least a 80486 personal computer but in two schools there were not enough 80486 computers to go around. Therefore the remaining 30 students had to take both Homework 1 and Homework 2 with paper and pencil. Although this was not in the initial plan, this arrangement served a useful purpose. By comparing these 30 students with those using computers to do their homework, the effect of using Electronic Homework could be tested.

The results of these students' homework completed either with Electronic Homework or paper and pencil, were recorded. Also, the learning results of doing homework, either by Electronic Homework or by conventional paper and pencil, were measured by a retention test consisting of items with difficulty levels similar to the homework. The retention tests were administered to the students approximately one week after the homework sessions. Together with the retention test, each student was asked to complete a Learning Processes Questionnaire. The purpose of this questionnaire was to categorize students as deep or surface learners. Lastly, students' academic abilities were calculated by using the following formula:

$$\text{ability score} = \frac{\text{student's mathematics score} - \text{class mean of mathematics score}}{\text{class standard deviation of mathematics score}} \times \text{standard deviation of homework score in the class}$$

where the mathematics scores were students' results in a recent examination or test as supplied by the respective schools, and the homework score was the score obtained in doing conventional homework. Since students came from different schools and thus might be of different standards, the use of this formula was to ensure that this score reflected their abilities. Exactly how this formula was obtained can be found in Chapter 5<sup>108</sup>.

**Effects on Homework Scores**

From the data obtained, students were categorized into three ability groups and two learning approach groups. Three ability groups, namely, high, median, and low groups and two learning approach groups, namely, the deep approach and the surface approach were thus identified. All scores can be found in Appendix M.

Scores obtained in the two homework exercises were compared by using multivariate analysis of variance with homework method (Electronic Homework or paper-and-pencil), academic ability (high, median or low), learning approaches (deep or surface) and school as the independent variables. Results of the analysis are shown in Table 8.1.

---

<sup>107</sup>Equivalent to US Grade 8.

<sup>108</sup>Page 166.

Table 8.1

(a)

Analysis of Variance for Homework Scores

Source of Variation	Sum of Squares	DF	F
Main Effects	1787.700	6	56.983***
Hwrktp	1515.628	2	144.933 ***
Acadtp	258.173	2	24.688***
Ltype	13.899	1	1.329

Note. Hwrktp = Homework Type; Acadtp = Academic Ability; Ltype = Learning Type.

\*:  $p < .05$ , \*\*:  $p < .01$ , \*\*\*  $p < .001$ .

(b) Mean Scores and cell sizes

	Hwrktp			Acadtp			Ltype		
	1	2	3	1	2	3	1	2	3
Sample size	50	65	28	50	50	43	7	88	48
Mean	44.19	51.07	64.04	43.69	49.00	62.50	43.45	52.26	50.41

Note. Hwrktp = Homework Type; Acadtp = Academic Ability; Ltype = Learning Type.



Table 8.1 shows that there was a main effect due to method of doing homework as represented by the variable Hwrktp in the analysis. The highest mean came from the group with Hwrktp=3, which was the group of students who did not use Electronic Homework in both their assignments. A cursory look at the figures seems to show that Electronic Homework is less effective than traditional homework methods. However, observation of homework sessions and perusal of data collected in this research, point to the possibility that the difference between the two groups resulted, not because the students were unable to solve many of the problems with their computers but because these students simply did not have enough time to finish the exercise. There could be three reasons for this.

1. Adaptation required. It was the first time any of the students had used a computer to do their homework. Not all of them found it easy to use mouse pointers or keyboards to enter their expressions. Most of them would need time to adjust and adapt themselves to this new homework environment. It is true that using a computer keyboard or a mouse as input device is generally much slower than using paper and pencil. In addition, it was found the Electronic Homework sometimes reacted quite slowly. All these factors together meant that students using Electronic Homework required more time to complete the same amount of work as those using the traditional method. As it turned out, only four of the participating students finished their assignments within the time limit while quite a number of students using conventional methods were able to complete the exercise although not all of them got these problems correct. The situation was even worse for some students who experienced computer failures.

2. Unfamiliar interface. The computer interface was new to some students, especially when it was used as a tool to do their homework. Some would make type-errors or wrong clicking which then required extra time to correct. On the other hand, students using traditional methods would not have this kind of problem.
3. Remediation strategy of Electronic Homework. The strategy used by Electronic Homework to help students correct their errors is as follows: Whenever an error occurs, the student has to discontinue the problem solving process to correct the error. Only when the error is corrected can he or she can continue with the remaining part of the problem. In conventional homework exercise, an error may be overlooked. Hence a completed exercise is not necessarily a correct one.

An overlooked error may eventually become a stable error. Hence, even though a student can finish more problems within a prespecified time, it does not mean that this student learns better than others. The homework result thus cannot be an indicator of the learning effect. This has to be measured by a later test (the retention test). The following sections describe the effect of using Electronic Homework on the retention test scores.

### **Effect of Electronic Homework on Retention Test Scores**

Although there were difficulties found in both the hardware and software during the experimental stage, there is evidence to show that Electronic Homework might in some way work as expected. This is based on the observation made during the evaluation process. Some students using Electronic Homework were mentally stimulated when the computer pointed out their errors and would do their best to



correct them. Although the actual effect of using the computer system will not be found until later analyses, this seems to be one additional advantage that was not assumed when the system was designed. Highly motivated students might achieve better results from Electronic Homework than from traditional methods. It should be noted that when the system was being tested, it was not equipped with any pictorial or sound effects which many developers of computer-assisted instructional system considered as essential. Therefore the students' motivation should not be related to any kind of extrinsic reward, but rather, it seems to come from intrinsic factors such as their desire to master the materials. This issue of motivation deserves further exploration and will be discussed again with the data obtained in the summative evaluation.

### **Effects on All the Participated Schools**

The effects of using Electronic Homework were investigated by analyzing the retention test scores which were obtained approximately one week after the homework exercise which include both the electronic and conventional methods, was done. The retention test consisted of two parts, referred to as part 1 and part 2 respectively. While items in part 1 are made equivalent to those which appeared in Homework 1, items in part 2 are equivalent to that in Homework 2. The scores of part 1 and part 2 were calculated separately as Rtp1 and Rtp2 respectively. The total score (Rttot) was also calculated by adding up Rtp1 and Rtp2. All three scores Rtp1, Rtp2 and Rttot were analyzed by using multivariate analysis of variance with Homework Method Type (Hwrktp), Academic Abilities Type (Acadtp), School Code (Schcode) and Learning Approach Type (Ltype) as the independent variables. No statistically significant effects could be found for any one of the dependent variables.



### **Transfer between Homework 1 and Homework 2**

It should be noted that problems in Homework 2, and those in part 2 of the retention test, are considered to be more abstract than problems in Homework 1 and part 1 of the retention test. The design of the present evaluation process was based on an assumption that if Electronic Homework is more helpful to students in doing less abstract problems, then students using Electronic Homework in Homework 1 would do better in the retention test, particularly in the part 1. On the other hand, if Electronic Homework can help students to do better in more abstract problems, the effect will be reversed. However, this assumption overlooked the possibility that although problems in Homework 1 and Homework 2 are different in terms of abstractness, the solutions of both types of problems actually require the use of the same set of axioms. Knowledge learned in one homework may thus be transferred to the other. Although there may be different degrees of transfer between that from Homework 1 to Homework 2 and that from Homework 2 to Homework 1, the occurrence of transfer would to a certain degrees counter-balance the effect of two different methods or work. The fact that there was no significant difference found between the homework types may be attributed to this although no evidence can be found from the data collected in this study.

### **Comparison between students who used Electronic Homework and those who did not**

The fact that 30 students from two of the participating schools did not have the chance to use Electronic Homework provided a chance to study the effect of the system on students who used it and on those who did not used it. As these students

were only randomly selected from the two schools (schcode=5 and schcode=6) with the lowest mean scores among the six schools, it would not be appropriate to compare the scores of these students to those of students in other schools. A new variable called Ehyn was then created to differentiate students who used Electronic Homework in their homework sessions from those who did not. The effect on the retention test scores by Ehyn was then compared together with other independent variables Acadtp and Ltype by using analysis of variance. For the same reasons, the analysis was done only within schools with schcode 5 and 6. Again, no significant difference could be found.

The above analysis seems to support the view that no difference that could be attributed to the homework methods. However, as this analysis was carried out only with students in two schools (schcode=5 and schcode=6), and, it happened that these were students of relatively low academic ability, it can still be argued that with students in other academic ability groups or other schools, the effect might be different. The following findings reveal such a possibility.

Table 8.2

Table showing the Mean Retention Test Scores of the Participated Schools

Schcode	Rtp1			Rtp2			Rttot		
	Mean	SD	Cases	Mean	SD	Cases	Mean	SD	Cases
1	7.62	3.03	35	3.69	3.15	35	11.31	5.23	35
2	10.56	1.68	36	7.22	2.23	36	17.78	3.51	36
3	8.82	3.80	29	4.86	3.36	29	13.69	6.60	29
4	7.81	3.00	36	4.53	3.52	36	12.33	6.12	36
5	5.83	3.34	36	3.17	3.22	36	9.00	6.04	36
6	6.89	2.73	28	2.57	2.12	28	9.46	4.20	28
Overall	7.94	3.31	200	3.40	3.32	200	12.33	6.11	200

Note. Rtp1= score in part one of retention; Rtp2=score in part two of retention test,  
Rttot=Total score in retention test.



### Effects on Individual Schools

Table 8.3 shows the results of analysis of variance for the total retention test score. Although the main effect attributed to Homework type was not significant, there were main effects which could be attributed to the two variables Schcode and Acadtp, which meant that students with different academic abilities or who belonged to different schools would have different retention scores. The effect of academic ability on the test scores would be generally agreed. However, further investigations should be carried out on the effect of school. Since the effects attributed to academic ability and learning process types had already been removed from the analysis, this school effect should, theoretically, come from some factor that was not measured in the present study. It is possible that there are some unobserved factors that characterize different schools and these factors were affecting the result of using the computer system. By studying the schools individually, it may be possible that the effect of using Electronic Homework can be detected. The following sections describe the results of such analyses.

Table 8.3

Analysis of Variance for the Retention Test Scores (Rttot) (N=220)

Source	Sum of Squares	DF	F
Schcode	1630.55	5	13.13***
Ltype	35.93	2	.72
Acadtp	719.84	2	14.50***
Hwrktp	9.01	1	.37

Note. Rttot=Total retention test score; Hwrktp = Homework type; Acadtp = Academic type; Ltype = Learning approach type.

\* p< .05. \*\* p<.01. \*\*\*p<.001

**School Effect**

Retention test scores of individual schools were analysed by using analysis of variance. In doing these analyses, two modifications were made in order to avoid statistical difficulties caused by empty cells. The first modification was due to fact that only 13 students could be categorized as using the deep learning approach. When these students were further divided into the different cells when two additional variables Academic Ability Type and Homework Type were added, some of these cells became empty and thus caused difficulties in the analyzing process. The variable Ltype was thus excluded in later analyses. Reasons why there were too few deep approach students might be interesting but are not be the focus of the present study.

The second modification was caused by the analysis to the students in a particular school (Schcode=2). The retention test scores of students in this school were so high that only two students could be found in the low academic ability group. In addition, these two students, by chance, were both in the group with Hwrktp equals 2. This caused the cell with Acadtp equals 1 and Hwrktp equals 1 empty. For this school, the analysis was done for the mediate and high academic ability groups only.

Of the six participated schools, significant results can only be found in one school (Schcode=2). Table 8.4 shows the results of analysis of variance for retention test scores for this school.



Table 8.4

Analysis of Variance for the retention test score with Sch=2 (N=35).

Source	Rtp1			Rtp2			Rttot		
	DF	SS	F	DF	SS	F	DF	SS	F
Hwrktp (H)	1	6.82	4.10*	1	1.34	.566	1	2.11	.26
Acadtp (A)	1	15.08	9.07**	1	37.10	9.31**	1	99.48	12.34***
H X A	1	7.46	4.49*	1	.11	-.03	1	9.41	1.17
Error	31	51.54		31	123.52		31	249.96	
Total	34	85.89		34	163.54		34	382.97	

Note. Hwrktp = Homework type; Acadtp = Academic type; Sch = School; Ltype = Learning approach type.

\*  $p < .05$ . \*\*  $p < .01$ . \*\*\* $p < .001$

Table 8.4 shows that students' academic abilities significantly affected all three scores in the retention test but the variable Hwrktp affected only the scores of Rtp1. As Rtp1 measures students' performances in doing problems in the first part of the retention test and that these problems were considered as less abstract than that in Rtp2. This result suggests that the use of Electronic Homework might help student on easier problems.

Interaction effect could also be found in the variable Rtp1. Post facto analysis showed that students with medium academic ability and used Electronic Homework to practise Homework 2 (abstract problems) scored even lower than that caused by the two main effects.

### **Who benefited more and who benefited less**

No significant effect could be found for the use of Electronic Homework in general. However, it is found that when students from two low academic ability schools were compared, the use of Electronic Homework has no effect. On the other hand, when the school with the highest academic ability was studied, it was found that Electronic Homework can help students to learn less abstract problems better. It is possible that only those students with high academic abilities would learn better when the system is used in less abstract problems. However, this was not supported by the analysis of the whole population.

The answer to this seems to lie in the differences among the schools. Exactly what caused the difference is not known, but it should not be anything related to students' computer literacy. According to the analysis done on students' perception on using Electronic Homework (Appendix L), there was no significant difference among

the schools on the number of students who owned computers and on the frequencies they use computers.

There can be two possible reasons that makes the students in the school that Electronic Homework has effects distinct from the others. The first one is that most of the students of this school came from higher socio-economic families. The second one, as reported earlier in this chapter, is that the students were highly motivated. Whether this socio-economic background or motivation can be the factors affecting the effect of using Electronic Homework has to be further investigated.

### **Students' Perception of Electronic Homework**

Students' perception of using Electronic Homework was collected (Appendix L) in the hope of finding out how students would like Electronic Homework to work for them. As there are 22 questions in the questionnaire and scores of most of them lie around the mean score 3, these questions were ranked and only the 5 questions with the highest scores and 5 of the lowest scores were reported. Table 8.5 shows these questions with their ranks and scores.

Information gathered from this analysis can be divided into two categories: The first one is on their use of computers. In general, the students felt no difficulty in using keyboard and mouse as the input devices, and they would like picture and sound effects after correct answers. Also, if required, they would buy a computer for use at home.



Table 8.5

Analysis of Students' Perception of Electronic Homework

Rank	Question No.	Question	Mean Score
1	Q19	The computer was slow to respond to my input.	3.88
2	Q22	If required, I can purchase a computer at home	3.79
3	Q6	I have no difficulty in using keyboard as input device.	3.71
4	Q5	I have no difficulty in using mouse as input device.	3.60
5	Q9	If an interesting picture or animation shows after a correct answer, I would be more motivated to learn.	3.22
18	Q3	The extra problems generated by the computer help me to consolidate the correct knowledge learned.	2.74
19	Q1	I can have constant feedback in the working process.	2.63
20	Q15	EH is interesting.	2.55
21	Q20	I would recommend EH to my classmates	2.49
22	Q8	I prefer using computer input method to do my homework.	2.38

The other category is on their perception on Electronic Homework. The general idea is that they would not like using a computer to do their homework. The reason may be that the system is not interesting (no picture or animation, too slow, etc.). A more detailed discussion of this will be given at the end of this chapter.

### **Teachers' Perception of Electronic Homework**

Teachers' perception on using Electronic Homework was also collected by using a questionnaire (Appendix J). Analysis of the results can be found in Appendix Q. Table 8.6 summarizes the 5 highest and 5 lowest rank order questions.

The teachers' perception on the use of computers as well as the use of Electronic Homework generally agreed with that of the students. They would prefer interesting pictures and beautiful music after correct answers. They also thought that the students would have no difficulty in using the mouse and keyboard as input devices. On the general idea of Electronic Homework, they admired the designing principle, but were reserved on the practical use of Electronic Homework. Considering the various constraints that Electronic Homework is now suffering, this attitude is reasonable. It should be pointed out that although the questions Q2, Q18 and Q20 (Table 8.6) were ranked quite low, the scores were all 3.00. The ranks were actually not too low.

Table 8.6

Analysis of Teachers' Perception on using Electronic Homework

Rank	Question No.	Question	Mean Score
1	Q5	There should be no difficulty for my students to use mouse as input device.	4.17
1	Q9	If a piece of music follows a correct answer, the students would be more motivated to learn.	4.17
3	Q8	If an interesting picture or animation shows after a correct answer, the student would be motivated to learn.	4.00
3	Q21	The idea of Electronic Homework should be helpful in general.	4.00
5	Q10	The individual report helps me to understand individual students' errors.	3.80
5	Q16	The group report saves a lot of my time in correcting students' exercise books.	3.80
5	Q22	I hope Electronic Homework can include other mathematics topics.	3.80
19	Q2	The feedback is helpful in correcting students' errors	3.00
19	Q18	Students can better master the materials by using Electronic Homework.	3.00
19	Q20	Students can work at their own pace by using Electronic Homework.	3.00
22	Q13	The report saves a lot of my time.	2.67
22	Q23	I would recommend it to other classes/teachers.	2.67



## Discussion

The evaluation process of Electronic Homework was full of difficulties that required tremendous effort to overcome. Some of the difficulties were solved but some remained. The following section summarizes all these difficulties and the implications. Suggestions are then made for the improvement of the system.

### Who Benefit From Using The System

A major problem of Electronic Homework is its slow response. Although for most of the expressions entered, the response time is acceptable, the occasional occurrence of a slow response would be enough to keep the student from using the software. Also, when the system is not equipped with attractive pictures and music, there seems no reason to keep the students in front of the computer. Only two types of students may benefit from using the system. The first type is those students with high academic ability who would make fewer unpredictable (by the computer) errors. These students would not have to wait very long since the computer could react reasonably fast to the correct expressions or predictable errors. The second type is those students who are highly motivated, either because they are anxious to use computers in learning or because they are challenged by difficult problems. These students would have the patience required to wait for the response from the computer even if it takes a long time. Results of the evaluation seem to support this assumption.

## **Hardware Constraints**

Even though Electronic Homework can help some students, its use is quite limited when it cannot help the others. As discussed above, Electronic Homework can be made to handle much more unpredictable errors. However, the computers now available to most students are so slow that sometimes they react very slowly when extensive searching among its rule base is required. This problem of speed was solved by not allowing the computer to compose rules during checking students' expressions, but instead, adding composite rules in the computer system before it was used. Since the quantity of the composite rules that can be added is limited, the system can now react faster to relatively frequent errors, but it loses its power to handle relatively rare errors. The solution to this problem has to wait until faster computers are available to our students. It may take one or two years for this to happen.

## **Human-computer interface**

If it is true that the current version of Electronic Homework is applicable to highly motivated students only, the improvement of the human-computer interface would surely help to motivate other students. The following sections describe how this interface can be improved.

### **Computer Language**

Electronic Homework used a computer language called Visual Basic to develop its human-computer interface. The language is quite sophisticated but there are still some bugs. There were cases that the interface was not stable so that students

had to restart every thing again. The improvement of Electronic Homework would thus in a way depend on a newer version of Visual Basic or depend on whether there is another computer language that can do a better job.

### **Input Devices**

Both teachers and students claimed that the students would not have difficulties in using the mouse pointers of keyboards as the input devices. However, the input method currently used by Electronic Homework is not as convenient as the traditional paper-and-pencil method. The effect of use of Electronic Homework could possibly be much improved if a better input method like voice input could be incorporated.

### **Sound and Pictures**

As summarized in Table 8.5 and Table 8.6, both the teachers and the students would prefer the display of pictures or animation or the produce of music after a correct answer. Although the benefit of such extrinsic reinforcer is doubtful since some students might find these annoying. However, at least the students should be allowed to choose whether they would like such effects while working process.

### **Effect on the use of Electronic Homework**

The evaluation process used in this study was not able to identify the exact reason why the effect of Electronic Homework could only be found in one school. This suggests that the process itself was not comprehensive enough. There were three factors which were not considered in sufficient detail in the present study viz. student



motivation, the transfer effect and the time limit. Each of these factors is discussed separately in the following paragraphs.

### **Students' Motivation**

Although it was suspected that the effect of using Electronic Homework was related to students' motivation, no attempt was made to measure this factor in the present study. In future studies, this should be included.

### **Transfer Effect**

A major assumption in the present study is that students would receive more help from Electronic Homework in less abstract problems. Hence, the evaluation was carried out mostly on comparing the effect on problems of different degrees of abstractness. This assumption overlooked the possibility that what students learned in one homework assignment could be transferred to another. Most students learned with Electronic Homework in either Homework 1 or Homework 2. Hence even if Electronic Homework does bring results, they would be diluted by this transfer effect.

In the test, by pure accident, thirty students did not use Electronic Homework in any of the homework session. These students came from two schools with lower academic ability streaming. The fact that Electronic Homework could not help these students should not be implied that Electronic Homework could not help students with higher academic abilities. In future studies, a comparison between those who use computer systems and those who do not should be included in the evaluation process.

### **Time Limit**

The evaluation of Electronic Homework was done in two homework sessions. Each session consisted of two periods. This experience shows that students were not given enough time to explore and get accustomed to the system. Besides, it is clearly not possible for any teaching method to be effective in just two sessions. It is thus suggested that in future, evaluation should be carried out over a longer period.

In the present study, it was arranged for the students to complete the test homework assignments at school and not at home in order to avoid bias: it is commonly recognized that personal computers in the home are indicators of a better socio-economic family background which, under normal circumstances, make it possible for children of such families to achieve higher academic results at school. With the lowering of the price of computers, this factor would become less important. It is hoped that eventually all students will be able to use Electronic Homework in the home, thus, enabling the system to be evaluated in a real situation.

### **Expert-Novice Differences**

In modifying the system so that it can respond faster to students' inputs, it is found that the response can be made faster if two or more rules are composed as a new single rule. By adding more and more composite rules in the system, the system can react faster for more different students' inputs. Currently, the composite rules are hand-coded. However, if this can be done by machines, this computer system can actually be a model of a human tutor and the creating of composite rules would be the

simulation of the developing process from a novice to an expert. Such a model may be valuable to the study of human cognitive development.

## **Chapter Summary**

This chapter discussed the findings from the evaluation of Electronic Homework in schools in Hong Kong. It was found that although there were many unfavourable factors that reduced the effectiveness of Electronic Homework. There were some students who learned better with the system. The exact reasons for this cannot be deduced in the present study, but several suggestions were made for the improvement of the system. It is hoped that with these modifications and the advancement of computer technology, Electronic Homework can help a wider population of students.



## ***CHAPTER 9***

# **CONCLUSIONS AND DISCUSSION**

### **Overview**

This project started with the exploration of various means of incorporating knowledge into a computer system and ended with a final product called Electronic Homework. The study was an investigation into the various aspects of developing an intelligent tutoring system which was, at the same time, a system which could assist teachers in handling homework problems of their students. The results of this study may be divided into two categories.

On the theoretical basis for developing this system, several questions were raised and answered. On the practical aspects of incorporating the findings into a tutoring system, different techniques and different types of knowledge were employed to make the system possible. Finally the effectiveness of the system was evaluated. Suggestions for further improvement, were derived from the results of the evaluation. This chapter summarises the findings and discusses their implications.

## Theoretical aspects

In the developing of Electronic Homework, a major aspect of the project was the investigation into the human process involved in problem solving. Special emphasis was placed on this in the investigation by means of computer simulation, of the underlying causes of student errors. A new type of error, referred to as "misperceived errors", was proposed. This type of misperceived error cannot be explained by using other theories such as repair theory or misgeneralization. A model in terms of inherited default values of frames was also suggested<sup>109</sup> for the easy explanation of the misperceiving process.

Unlike the traditional classification of knowledge as procedural knowledge and declarative knowledge, the present study suggested that knowledge in a computer system which simulates the human problem solving process could also be categorized as explicit knowledge and implicit knowledge. Explicit knowledge is knowledge that is explicitly represented either in the form of rules or any other means. It can be reported or modified. On the other hand, implicit knowledge is knowledge that is not represented in a computer system, but is assumed when the system is applied. As it is possible to modify explicit knowledge in a computer system, several pieces of knowledge can be combined to form a new piece of knowledge. If pieces of knowledge are represented as rules, then the combined knowledge can be referred to as a composite rule. Composition of rules and the proceduralization of declarative knowledge into procedural knowledge are the two subprocesses involved in the

---

<sup>109</sup> Page 194.

process of skill acquisition (Anderson, 1990, 1993). However, in terms of the knowledge representation system used in the present study, there is no proceduralization of knowledge. Composition of rules is the sole function that can speed up a computer system's response.

The process of the composition of rules was simulated in Electronic Homework. Originally, the system consisted of basic correct and incorrect rules. When a student entered an expression that did not correspond to either one of these rules, the system tried to combine two or more rules to explain the expression. This process required considerable time. At a later stage of the development, the system was equipped with more composite rules and could then react much faster. This resembles the skill acquisition process suggested by Anderson (1990, 1993), but the system was able to simulate this process without any proceduralization of knowledge. The implication is that proceduralization may not be necessary in a system for explaining the process of skill acquisition.

The following sections attempt to combine the findings from the present study with existing theories to answer two questions that are frequently encountered. These questions are respectively why and how do students make errors and what makes an expert tutor?

### **Why and how do students make errors?**

There have been various suggestions on why and how students make errors during their problem solving processes. The two main approaches are the bug theory of Brown & Burton (1978) and the use of mal-rules (Young & O'Shea (1981);



Sleeman, 1984). The latter however is only a way of representing errors rather than a way of explaining why and how errors occur. Some other theories of errors together with findings of the present study are briefly discussed below:

### **Repair Theory**

The first theory that can explain why and how errors occur may be the Repair Theory suggested by VanLehn (1982a) to account for errors by students during subtraction problems. According to VanLehn,

*When a student gets stuck while executing his possibly incomplete subtraction procedure, he is unlikely to just quit as a computer does when it can't execute the next step in a procedure. Instead, the student will do a small amount of problem-solving, just enough to get "unstuck" and complete the subtraction problem. These local problem-solving strategies are called "repairs" despite the fact that they rarely succeed in rectifying the broken procedure. (VanLehn 1982a, 1982b)*

Hence error originates from an impasse and the error is caused by a problem solving process called Repair. This impasse-repair pair then becomes a rule which causes subsequent errors when identical situations recur.

### **Misgeneralization**

While the repair process is referred to by VanLehn as “task-general methods”, most of which are familiar to most subjects” (VanLehn, 1990)<sup>110</sup>, other researchers have suggested that errors are caused by more domain-specific knowledge. Matz

---

<sup>110</sup>Refer to page 95 of Chapter 3 for more details.

(1982) has suggested that errors are the results of reasonable, although unsuccessful, attempts to adapt previously acquired knowledge to a new situation. These kinds of errors are said to be caused by misgeneralization.

### **Errors without Impasse**

The above two explanations for errors have one thing in common - they are both impasse-driven. In both explanations, a student has to experience a situation which he or she does not know how to resolve. The student then employs either a “task-general” method (VanLehn, 1990) or a domain-specific method (misgeneralization as given by Matz, 1982) to overcome the difficulty. It is possible that some observed errors may be explained by these impasse-driven theories. On the other hand, there may be some other errors which these theories can not explain adequately.

An example of errors that are not driven by impasses are those errors that were the correct responses to related problems (Norem & Knight, 1930)<sup>111</sup>. Payne & Squibb (1990) also suggested errors may also be caused by rules induced from the written input and output of productions<sup>112</sup>. In both cases, subjects may not experience any kind of impasse but directly use the correct rules, be they the responses to related problems or those induced from external display. Impasses are then not necessary conditions for any errors to occur.

---

<sup>111</sup>Refer to page 100 for more details.

<sup>112</sup>Refer to page 103 for more details.

## **Misperception**

It is argued in Chapter 3<sup>113</sup> that errors may also be caused because students misperceive the problem expression then use a correct rule for the perceived situation. The two types of errors discussed above: errors caused by using correct rules in related problems (Norem & Knight, 1930) and errors caused by external display (Payne & Squibb, 1990), are actually describing the different stages of misperception. Subjects may misperceive the problem situation, either because of the external display or other means, he or she then use correct rules that are correct in the perceived situation to solve the problem, an error occurs.

An error caused by misperception is characterized by its automaticity and fast speed since the student does not experience any kind of impasse. The error is generated by directly applying a rule. In the present study, there was evidence to show that some students did make this kind of misperceived errors. A simple example is that a student expressed  $0.4771-4.771$  as  $0.4771/4.771$  due to a misperception that the first expression is an expression consists of logarithms<sup>114</sup>.

Misperceived errors can be explained more clearly by using frames with default values<sup>115</sup>. Knowledge perceived from a problem representation can be thought to be stored in the slots of a frame. If, for some reason, some knowledge pieces that should be perceived from the problem representation were missed, slots for these knowledge pieces will be filled by some default values. The problem is thus mis-

---

<sup>113</sup>Page 105.

<sup>114</sup>Page 194.

<sup>115</sup>Refer to page 194 for more details.



represented and rules applicable to this mis-represented problem are then applied. An error then occurs.

### **Meta-rules for Errors**

The fact that some errors are caused by misperception does not imply that all errors are caused in this way. As both misperception and misgeneralization are internal processes and are not easily separated one from the other, it is also not easy to say whether an error is caused by misgeneralization or misperception. Further, although misperception and misgeneralization are two different processes, the results of both processes can be represented by rules. These rules should be distinguished from other mal-rules since one such rule can generate many mal-rules<sup>116</sup>. These rules are referred to as meta-rules in the present study. If we ignore the differences between the internal process involved in misperception and misgeneralization but focus on what these meta-rules are, it may be easier for us to find ways of helping students to correct their errors.

Rules that representing reasons that errors occur were referred to as the meta-bugs by Giangrandi & Tasso (1995). Five such rules, referred to as the meta-rules, were also identified in this study<sup>117</sup>. When these meta-rules were incorporated into Electronic Homework, it was found that 90.17% of the mal-rules identified could be explained by using one or two of the meta-rules in addition to the correct rules.

---

<sup>116</sup>In chapter 6, students errors in doing logarithm problems are categorized. One category corresponds to one such rule. Refer to page 216 for more details.

<sup>117</sup>Page 206.

### **Implications for the design of Electronic Homework**

The identification of meta-rules has two implications. First, it points out that although there are many possible errors, each of which can be represented by a mal-rule, they are caused by just a few meta-rules (either because of misperceiving or misgeneralizing). If we could find out what these reasons are and try to help the students to correct them, these errors will be removed. This forms the underlying principle for the design of the present tutoring system, the Electronic Homework.

The second implication is that if errors can be explained by just a few meta-rules, then the system only needs to store these meta-rules instead of storing large numbers of mal-rules. The size of the system will then be much reduced without loss of any of its functions. It would be simple to expand the system to include more knowledge and can thus tutor a wider subject area.

The above two implications affect the tutoring strategies and the knowledge representation method used in Electronic Homework. Tutoring strategies employed in the system were obtained from human teachers, but as shown in Chapter 6, most of these tutoring strategies were found to be related to these meta-rules. On the problem of knowledge representation, the earlier version of Electronic Homework was designed that as few rules as possible were incorporated. But it was later found that this design was impractical since in some cases the system reacted slowly. More rules, including composite rules were then added.

The decision to add rules to the system was based on practical considerations and did not necessarily reflect any relationship between the computer system and the human process. However, when this was done and the performance of the system was

observed, a lot of resemblances appeared. The following section summarizes the findings concerning this human-machine resemblance.

### **What makes an expert tutor?**

It was discussed in Chapter 2 that two or more rules can be composed into one so that the composed rule can achieve the same purpose as the original rule but functions faster. It was also discussed in the above sections and was revealed in Chapter 6 that by using meta-rules, fewer rules can be used to represent more errors in the computer system. An immediate question would be whether or not more rules, including the composed rules, should be incorporated into the computer system. On one hand, the adding of composed rules can make the system react faster to students' inputs. On the other hand, these rules would use up memory space and make it harder to incorporate more knowledge into the system. As Electronic Homework was originally designed to simulate a human tutor, it is worthwhile looking more closely at the question of whether human beings do store redundant rules in their minds.

### **Redundancy of Knowledge**

There are a lot of examples of redundancy of knowledge in the human brain. A simple example is that the knowledge "cod can swim" is both stored as an attribute of the concept "fish" and the concept "cod". Hence, if you want to verify the statement "a cod can swim", you can either go to the attributes of the concept "cod" to see whether there is an attribute "can swim", or you can go to the concept "cod" to see whether it is an instance of the concept "fish" and then go to the concept "fish" to see whether there is an attribute "swim".



In the context of mathematics, there are also a lot of examples. A simple one is when you first learn to do logarithm problems, you may be asked to simplify the expression:

$$[E71] \log 6$$

You would probably express it as:

$$[E72] \log(2*3)$$

and then express it as:

$$[E73] \log 2 + \log 3$$

However, an expert in logarithms would go directly from [E71] to [E73]. This does not mean that the expert does not store the knowledge required to express [E71] to [E72] and then [E72] to [E73]. In fact, the expert would store both the rules:

$$[R71] \log X = \log YZ \quad \text{where the product of } Y \text{ and } Z \text{ is } X$$

$$[R72] \log YZ = \log Y + \log Z$$

and the composed rule:

$$[R73] \log X = \log Y + \log Z \quad \text{where the product of } Y \text{ and } Z \text{ is } X$$

Redundancy of knowledge is thus not rare in the human mind.

### **Expert Tutor versus Novice Tutor**

The difference between an expert tutor and a novice tutor is that the expert can provide fast and accurate diagnosis and remediation, while the novice has to think for a long time before providing suitable help. When Electronic Homework was first developed, the system was mainly equipped with correct rules and meta-rules, but only with a very small number of mal-rules. The system was found to be too slow to provide useful information to students. The system was then revised by adding more

composed rules. It can now run much faster. In a way, the revised Electronic Homework is like the expert tutor while the older version is like the novice tutor.

The key to up-grading Electronic Homework to an expert tutor is thus by adding more composed rules. If Electronic Homework can be a model of the human tutor, then a human novice tutor may become an expert by adding more composed rules. Whether this is true or not is beyond the scope of the present study, but this might be an interesting research topic for future investigation.

## Knowledge obtained

In this study, text books, teachers and students provided various types of knowledge. Text book knowledge is how to correctly solve problems. Knowledge obtained from teachers included tutoring knowledge, estimation of problem complexity and estimation of problem difficulty. Knowledge obtained from students included mal-rules and their estimation of problem complexity. All the knowledge acquired was incorporated into the computer system. An interesting finding was one regarding the problem complexity. It was found that students could estimate problem difficulty more accurately than the teachers. For this reason, the calculations of problem difficulty were based on the students' estimation.

## **Can Electronic Homework Help Students And Teachers?**

The ultimate goal of the present study is to develop an intelligent tutoring system that can help students with their homework and provide teachers with a tool to help teachers to administer homework. The emphasis is placed on the tutoring system since if it can be proved that the system can help students to learn better, it would, indirectly benefit teachers. All the theoretical work was aimed to develop an effective tutoring system. The final stage of the present study is therefore the evaluation of the system.

### **Purposes of the Evaluation**

Electronic Homework is now only in a primitive form. Many features like sound and pictorial effect are not incorporated, and thus it is not expected that the system would help all students. However, it is in this primitive form that the effectiveness of the underlying principle can truly be tested. The reason is that if there is an effect, then it could only be attributed to the designing principles such as the tutoring strategies or the ordering of problems. Besides, through the evaluation of this primitive form, opinions of both teachers and teachers can be collected for future improvement of the system. Hence, the evaluation of the system is intended to find out the answers to two questions: whether the design can help students and what can be done further to help the students?.



## **Results of The Evaluation**

Answers to these questions were found in the results of the evaluation. Firstly, the system, even without visual or sound effects to arouse interest, was found to be helpful to a particular type of students - those who were considered to be highly motivated towards using computer systems (and maybe other learning tools too), although further evidence is needed to justify this assertion. Both the survey on the teachers' and students' opinions showed that they all preferred the incorporation of pictorial and sound effects. This points to the direction for future development.

The evaluation process also revealed several constraints on the use of Electronic Homework. Currently the computers commonly used in schools of Hong Kong are not sophisticated enough to handle at an acceptable speed the complex tasks required by Electronic Homework. Besides, the computer language Visual Basic and maybe the Chinese Windows are unstable in some cases. Both problems have to be solved before Electronic Homework can be used as a practical complement to classroom teaching.

## **Suggestions**

A major difference between a human tutor and a computer tutor system such as Electronic Homework is that a human tutor can learn but the computer tutor cannot. Initially a novice human tutor may not know how to diagnose students' errors nor how to remedy them. With increasing knowledge and experience, he or she would eventually become a better teacher. A computer tutor does not have this capability. Currently the rules in Electronic Homework are all hand-coded and then incorporated. The

computer system would never “grow up” until further human knowledge can be entered.

### Machine Learning

For computer tutors designed to take care of smaller subject areas, it is still possible to code the knowledge by hand. For larger systems, this is extremely difficult. Some kind of machine learning techniques should be incorporated into the system. In this case, the system could be trained using given examples and then automatically generate the rules. Much effort required to develop the system would be reduced.

### Input Systems

The system can also be improved with several new technologies besides the sound and pictorial effects. For example, a phonetic input system can allow students to enter their expressions orally. Hand-writers input systems can allow them to enter as they usually do in their homework assignments. At present, these two technologies may not be good enough to handle the complex input task for the present system. With time, oral or written input would become reality.

### Better understanding of Human Problem Solving

#### Process

Electronic Homework suffers from its slow reaction and low diagnosing power. However, one of these can be improved at the expense of the other. For example, the speed can be made faster by deleting some of the rules in the system, but

at the same time, the number of undiagnosed error will be increased. On the other hand, more rules can be added to the system to enable the system to diagnose more errors. But the system will require more time to search for the right rule, which will result in a slower response. Only when we have more powerful computers can we increase the performance in both or these dimensions.

However, with human tutors, there is no need to sacrifice one aspect to improve on the other. On the contrary, an expert tutor is able to quickly diagnose common errors but at the same time, diagnose more rare errors in a shorter time than novice tutors do. The increase in knowledge to diagnose common errors does not result in lowering of their diagnosing abilities. Considering the processing speed of the human brain when compared with computers, this would suggest that human beings can use a more efficient strategy to handle this diagnosing problem while the computer can only scan from the start to the end of its rule base.

A possible answer to the question why human tutors can out-perform machine tutors is that human tutors use meta-knowledge, which is the knowledge of how to use the knowledge stored. Electronic Homework uses same meta-rules to represent the rules that explain mal-rules. It may be possible that further exploration of human knowledge might help to identify this kind of meta-knowledge and incorporate it into the computer system. The performance of such a system would then be comparable to that of a human tutor.



## References

- Amzi! (1996). Manual of Amzi! Prolog Ver.3.1. Massachusetts: Amzi! inc.
- Anderson, J. R. (1983). The architecture of cognition. Cambridge, MA: Harvard University Press.
- Anderson, J. R. (1990). Cognitive psychology and its implications (3rd ed.). New York: W. H. Freeman.
- Anderson, J. R. (1992). Intelligent tutoring and high school mathematics. In C. Frasson, G. Gauthier & G. I. McCalla (Eds), Proceedings, Intelligent tutoring systems, Second international conference, ITS '92. Springer-Verlag.
- Anderson, J. R., (1993). Rules of the mind. Hillsdale, NJ: Lawrence Erlbaum.
- Anderson, J. R., Boyle, C. E., & Yost, G. (1985). The geometry tutor. Proceedings of the International Joint Conference on Artificial Intelligence, Los Angeles, pp. 1-7.
- Anderson J. R., Corbett, A. T., Koedinger, K. R., & Pelletier, R. (1995). Cognitive tutors: Lessons learned. The journal of learning sciences. In press.
- Anderson, J. R., & Reiser, B. J. (1985). The LISP tutor. Byte, pp. 159-175.
- Bechtel, W., & Abrahamsen, A. (1991). Connectionism and the mind: An introduction to parallel processing in networks. Basil Blackwell.
- Biggs, J. B. (1992). Why and how do Hong Kong students learn? Using the learning and study process questionnaires. Education papers. Faculty of Education. University of Hong Kong.
- Brown, J. S., & Burton, R. R. (1978). Diagnostic Models for procedural bugs in basic mathematical skills. Cognitive Science, 2, 155-192.

- Bunderson, V. C., & Olsen, J. B. (1983). Mental errors in arithmetic skills: Their diagnosis in precollege students. (Final project report, NSF SED 80-12500). Prove, UT: WICAT Education Institution.
- Burton, R. R., & Brown, J. S. (1979). An investigation of computer coaching for informal learning activities. *International journal of man-machine studies*, 11, pp. 5-24.
- Burton, R. B. (1981). DEBUGGY: Diagnosis of errors in basic mathematics skills. In D. H. Sleeman & J. S. Brown (Eds). *Intelligent Tutoring System*. London: Academic Press.
- Burton, R. R., & Brown, J. S. (1982). An investigation of computer coaching for informal learning activities. In D. H. Sleeman & J. S. Brown (Eds). *Intelligent Tutoring System*. London: Academic Press.
- Campbell, J. I. D. & Graham, D. J. (1985). Mental multiplication skill: Structure, process and acquisition. *Canadian journal of psychology* 39, 338-366.
- Cauzinille-Marneche, E. & Mathieu, J. (1988). Experimental data for the design of a microworld-based system for algebra. In Mandl, H, & Lesgold, A. (Eds). *Learning issues for intelligent tutoring system*. New York: Springer-Verlag.
- Chabris, C. F. (1991). *Artificial intelligence & Turbo C*. Tokyo: Toppan.
- Chan, T., & Baskin, A. B. (1990). Learning companion systems. In Frasson, C., & Gauthier, G. (Eds). *Intelligent tutoring systems: At the crossroad of artificial intelligence and education*. Norwook, NJ: Ablex.
- Chan, T., Chee, Y. S, & Lim, E. L. (1992). Cognition: An extended computational theory of cognition. In Frasson, C., Gauthier, G., & McCalla, G. I., (Eds).

- Intelligent tutoring systems: Proceedings of Second international conference, ITS '92. Berlin Heidelberg: Springer-verlag.
- Clancey, W.J. (1988). The role of qualitative models in instruction. In J. Self (Eds), Artificial intelligence and human learning. Chapman and Hall.
- Clocksin, W. F., & Mellish, C. S. (1987). Programming in Prolog. Berlin Heidelberg: Springer-Verlag.
- Cohen, N. J., & Squire, L. R. (1980). Preserved learning and retention of pattern analyzing skills in amnesia: Dissociation of knowing how and knowing that. Science, 210, pp. 207-210.
- Corbett, A. T., & Anderson, J. R. (1992). LISP intelligent tutoring system: Research in skill acquisition. In Larkin, J. H. & Chabay, R. W. (Eds). Computer-assisted instruction and intelligent tutoring systems: Shared goals and complementary approaches. Hillsdale, NJ: Lawrence Erlbaum Associates.
- Corkin, S. (1968). Acquisition of motor skill after bilateral medial temporal excision. Neuropsychologia, 6, pp. 255-265.
- Cox, R., & Brna, P. (1995). Supporting the use of external representations in problem solving: The need for flexible learning environments. Journal of artificial intelligence in education. 6 (2/3), pp. 239-302.
- Cumming, G. & Self, J (1991). Learner modeling in collaborative intelligent educational systems. In Goodyear. P (Eds). Teaching knowledge and intelligent tutoring. NJ: Ablex.
- Davis, R. B. (1984). Learning mathematics: the cognitive science approach to mathematics education. NY. Ablex.



## References

- Douglas, S. A. (1991). Tutoring as interaction: detecting and repairing tutoring failures. In Goodyear, P. (Eds). Teaching knowledge and intelligent tutoring. NY: Ablex.
- Drago, P. (1993). Teaching with spreadsheets: An example from heat transfer. Physics teacher, V31, n5, pp. 316-17.
- Education Time Sourceware (1995). Mathematics Review. Acton, MA: William K. Bradford.
- Elsom-cook, M. (1988). Guided discovery tutoring and bounded user modeling. In J. Self (Eds), Artificial intelligence and human learning. Chapman and Hall.
- Fox, B. A. (1991). Cognitive and interactional aspects of correction in tutoring. In Goodyear, P. (Eds). Teaching knowledge and intelligent tutoring. NY: Ablex.
- Gagné, E. H. (1985). The cognitive psychology of school learning. Boston Toronto: Little Brown & Co.
- Gentner, D., & Gentner, D. R. (1982). Flowing waters or teeming crowds: Mental models of electricity. In D. Gentner & A. L. Stevens(Eds.), Mental models. pp. 99-129. Hillsdale, NJ: Lawrence Erlbaum Assoc.
- Giangrandi, P. & Tasso, C. (1995). Truth maintenance techniques for modelling student's behaviour. Journal of artificial intelligence in education, 6, No. 2/3. pp. 153-202.
- Goodyear, P. (1991). Research on teaching and the design of tutoring systems. In Goodyear, P (Eds). Teaching knowledge and intelligent tutoring. NJ: Ablex.
- Gronlund, N. E. (1981). Measurement and evaluation in teaching. NY: Macmillan.

- Haugeland, J. (1978). The nature and plausibility of cognitivism. *The behavioral and brain sciences* 2: pp. 215-260.
- Hayes, P. J. (1979). The logic of frames. In Branchman, R. J. & Levesque H. J. (Eds) (1985). *Readings in knowledge representation*. San Mateo, California: Morgan Kaufmann.
- Hinsley, D., Hayes, J. R., & Simon, H. A. (1976). From words to equations: Meaning and representation in algebra word problems. In M. Just and F. Carpenter (Eds.), *Cognitive processes in comprehension*. NJ: Lawrence Erlbaum Assoc.
- Hornke, L. F., & Habon, M. W. (1986). Rule-based bank construction and evaluation within the linear logistic framework. *Applied psychological measurement*, 10(4), pp. 369-380.
- Houston, J. P. (1986). *Fundamentals of learning and memory* (3rd ed.). New York: Harcourt Brace Jovanovich.
- Jean-Francois, N. (1992). Reference network: A genetic model for intelligent tutoring systems. In Frasson, C., Gauthier, G., & McCalla, G. I., (Eds). *Intelligent tutoring systems: Proceedings of Second international conference, ITS '92*. Berlin Heidelberg: Springer-verlag.
- Jerman, M. E. (1983). Problem length as a structural variable in verbal in verbal arithmetic problems. *Educational studies in mathematics*, 5, 109-123.
- Johnson, D. C., Cox, M. j., & Watson, D. M. (1994). Evaluating the impact of IT on pupils' achievements. *Journal of computer assisted learning* 10, pp. 138-156.
- Kalat, J. W. (1984). *Biological psychology*. Belmont, Calif.: Wadsworth.

- Kamsteeg, P; & Bierman, D. (1991). Teacher thinking, learner thinking: Thinking-aloud protocol analysis in tutoring systems research. In Goodyear, P. (Eds). Teaching knowledge and intelligent tutoring. NY: Ablex.
- Kono, Y; Ikeda, M; & Mizoguchi, R. (1992). To contradict is human -- student modeling of inconsistency. In Frasson, C., Gauthier, G., % McCalla, G. I., (Eds). Intelligent tutoring systems: Proceedings of Second international conference, ITS '92. Berlin Heidelberg: Springer-verlag.
- Knowledge Revolution (1994). Interactive Physics. San Mateo, CA: Knowledge Revolution.
- Lai, M. Y. B. (1995). The attitudes of secondary students toward computer assisted instruction. Unpublished Report. The Chinese University of Hong Kong.
- Laird, J. E., Newell, A., & Rosenbloom, P. S. (1987). Soar: An architecture for general intelligence. Artificial Intelligence, 33, pp. 1-64.
- Larkin, J. H. (1989). display-based problem solving. In Klahr D., & Kotovsky, K. (Eds), Complex information processing: The impact of Herbert A. Simon. Hillsdale, NJ: Erlbaum.
- Lee, F. L. (1981). Analysis of cognitive strategies of problem solving process in mathematics and physics. Master Thesis, The Chinese University of Hong Kong.
- Lee, F. L. (1992). CP system. Unpublished Report. The Chinese University of Hong Kong.
- Lee, F. L. (1994). Slips and systematic errors: A comprehensive explanation based on harmony theory. Presented in the ninth annual conference of HKERA.



- Lee, F. L. (1995a). The rule of conceptual knowledge in remediation of procedural errors. Educational Journal, CUHK. Vol. 23, No. 1,
- Lee, F. L. (1996). Logex: A simple intelligent tutoring system. Educational Research Journal. In press.
- Lester, F. K. (1980). Problem solving: Is it a problem? In M. M. Lindquist (Ed.), Selected issues in mathematics education. pp. 29-45. Chicago: McCutchan.
- Leung, K. T. (1967). Elementary Set Theory. Hong Kong University Press.
- Lewis, M. W., Milson, R., & Anderson J. R. (1987). The teacher's apprentice: Designing an Intelligent authoring system for high school mathematics. In Kearsley, G. (Eds). Artificial intelligence and instruction: Applications and Methods. Addison-Wesley.
- Linville, W. J. (1970). The effects of syntax and vocabulary upon the difficulty of verbal arithmetic problems with fourth grade students. Dissertation Abstracts International, 30, 4310A.
- Liu, Y. (1992). Effects of computer-assisted instruction on cognitive outcomes: A meta-analysis. Journal of research on computing in education. Vol. 24. No. 3.
- Loftus, G. R. & Loftus, E. F. (1976). Human memory: The processing of information. New York: Erlbaum.
- Macnab, D. S., & Cummine, J. A. (1986). Teaching mathematics 11-16: A difficulty-centred approach. London: Basil Blackwell.
- Mandl, H, & Lesgold, A. (Eds) (1988). Learning issues for intelligent tutoring system. New York: Springer-Verlag.

- Marzano, R. J., & Jesse, D. M. (1987). A study of general cognitive operations in two achievement test batteries and their relationship to item difficulty. Washington, D. C.: Office of educational research and improvement, Department of Education. (ERIC Document reproduction service No. ED 299321).
- Mason, E., Zollman, A., Bramble, W. J., & O'Brien, J. (1992). Response time and item difficulty in a computer-based high school mathematics course. *Focus on Learning in Mathematics*. Vol. 14, No. 3. pp. 41-51.
- Mao, Y & Lin, J. (1992). Intelligent tutoring system for symbolic calculating. In C. Frasson, G. Gauthier & G. I. McCalla (Eds), *Proceedings, Intelligent tutoring systems, Second international conference, ITS '92*. Springer-Verlag.
- Marcke, K. V. (1992). Instructional Expertise. In Frasson, C., Gauthier, G., & McCalla, G. I., (Eds). *Intelligent tutoring systems: Proceedings of Second international conference, ITS '92*. Berlin Heidelberg: Springer-verlag.
- Martinak, R., Schneider, B., & Sleeman, D. (1987). A comparative analysis of approaches for correcting algebra errors via an intelligent tutoring system. *Proceedings of AERA, Washington, DC*.
- Matz, M. (1982). Towards a process model for high school algebra errors. In D. Sleeman & J. S. Brown, (Eds). *Intelligent tutoring systems*. New York: Academic Press.
- Mayer, R. E. (1975). Information processing variables in learning to solve problems. *Review of Educational Research*, 45, 525-541.
- McKendree J. E., & Anderson, J. R. (1987). Frequency and practice effects on the composition of knowledge in LISP evaluation. In J. M. Carroll (Ed.),

- Cognitive aspects of human-computer Interaction (pp. 236-259).  
Cambridge, MA: MIT Press.
- Minsky, M. (1981). A framework for representing knowledge. In Branchman, R. J.  
& Levesque H. J. (Eds) (1985). Readings in knowledge representation. San  
Mateo, California: Morgan Kaufmann.
- Moore, J. L., & Sleeman, D. (1988). Enhancing PIXIE's tutoring capabilities.  
International journal of man-machine studies, 28. pp. 605-623.
- MYTHOS Software Inc. (1993). Computer works. Software Marketing  
Corporation.
- Nehemie, P. (1992). A systemic approach for student modelling in a multi-agent  
aided learning environment. In Frasson, C., Gauthier, G., & McCalla, G. I.,  
(Eds). Intelligent tutoring systems: Proceedings of Second international  
conference, ITS '92. Berlin Heidelberg: Springer-verlag.
- Newman, D. L., Kundert, D. K., Lane, D. S., & Bull, K. S. (1988). Effect of varying  
item order on multiple-choice test scores: Importance of statistical and  
cognitive difficulty. Applied Measurement in education, 1(1), pp. 89-97.
- Newell, A. (1991). Unified theories of cognition. Cambridge, MA: Cambridge  
University Press.
- Newell, A., & Simon, H. A. (1972). The logic theory machine: A complex  
information processing system. I.R.E. Transaction of information theory.  
Vol. 17.2, No. 3, pp. 61-79.
- Newman, D. L., Kundert, D. K., Lane, D. S., & Bull, K. S. (1988). Effects of  
varying item orders on multiple-choice test scores: Importance of statistical  
and cognitive difficulty. Applied measurement in education 1 (1), pp. 89-97.



- Norem, G. M. & Knight, F.B. (1930). The learning of one hundred multiplication combinations. In: National society for the study of education: Report on the society's committee on arithmetic, Vol. 15, NSSE Yearbook 29, pp. 551-567.
- Ohlsson, S. (1987). Transfer of training in procedural learning: A matter of conjectures and refutations. In L. Bolc (Ed.). Computational models of learning. Springer-Verlag.
- Payne, S. J. (1988). Methods and mental models in theories of cognitive skill. In J. Self (Eds), Artificial intelligence and human learning. Chapman and Hall.
- Paolo, G. & Tasso, C. (1995). truth maintenance techniques for modelling student's behaviour. Journal of artificial intelligence in education. Vol. 6, No. 2/3. pp. 153-202.
- Park, O. (1991). Functional characteristics of intelligent computer-assisted instruction: Intelligent features. IN The Educational Technology Anthology Series: Expert Systems and Intelligent Computer-Aided Instruction. New Jersey: Eaglewood Cliffs.
- Park O., & Seidel R.J. (1991). Conventional CBI versus intelligent CAI: suggestions for the development of future system. In The Educational Technology Anthology Series: Expert Systems and Intelligent Computer-Aided Instruction. New Jersey; Eaglewood Cliffs.
- Payne, S. J. & Squibb, H. R. (1990). Algebra Mal-rules and cognitive accounts of error. Cognitive science 14, Pp. 445-481.
- Pennington, N., & Nicolich, R. (1991). Transfer of training between programming subtasks: Is knowledge really use specific? In J. Koenemann-Bellivean, T.

- G. Moher, & S. P., Robertson (Eds). Proceedings of the fourth workshop on empirical studies of programmers (pp. 156-176). Norwood, MJ: Albex.
- Plake, B. S., Glover, J. A., Kraft, R. G., & Dinnel, D. (1984). Cognitive capacity usages in responding to test items. Journal of Psychoeducational Assessment, 2 (4), 333-343.
- Polya, G. (1957). How to Solve it: A new aspect of mathematical method. Princeton, NJ: Princeton U Press.
- Resnick, L. B. (1982). Syntax and semantics in learning to subtract. In P. P. Carpenter, J. M. Moser & T. A. Romberg (Eds.), Addition and subtraction: A cognitive developmental perspective. pp. 136-155. Hillsdale, NJ: Lawrence Erlbaum Assoc.
- Resnick, L. (1984). Beyond error analysis: The role of understanding in elementary school arithmetic. Pittsburgh, PA: University of Pittsburgh, Learning Research and Development Center.
- Roberts, F.C., & Park, O. (1991). Intelligent computer-assisted instruction: An explanation and overview. IN The Educational Technology Anthology Series: Expert Systems and Intelligent Computer-Aided Instruction. New Jersey: Eaglewood Cliffs.
- Rumelhart, D. E., Hinton, G. E., & McClelland, J. L. (1986). A general framework for parallel distributed processing. In Rumelhart, D. E., & McClelland, J. L. (Eds). Parallel distributed processing: Explorations in the microstructure of cognition. Vol. 1: Foundations. MIT Press.

- Rumelhart, D. E., & McClelland, J. L. (Eds.). (1986). Parallel distributed processing: Explorations in the microstructure of cognition. Vol. 1: Foundations. MIT Press.
- Scandura, J. M. (1977). Problem solving: A structural/process approach with instructional implications. New York: Academic Press.
- Scientia, Inc. (1995). Advanced physics. Acton, MA: William K. Bradford.
- Searle, J. R. (1980). Minds, brains and programs. Behavioral and brain sciences, 3, pp. 417-424.
- Sebrechts, M. M. (1992). From testing to training-evaluating automated diagnosis in statistics and algebra. In Frasson, C., Gauthier, G., & McCalla, G. I., (Eds). Intelligent tutoring systems: Proceedings of Second international conference, ITS '92. Berlin Heidelberg: Springer-verlag.
- Self, J. (1988). Student models: what use are they? In P. Ercoli & R. Lewis (Eds). Artificial intelligence tools in education. North-Holland.
- Self, J. A. (1990). Bypassing the intractable problem of student modeling. In C. Frasson, G. Gauthier (Eds). Intelligent Tutoring Systems: At the crossroad of artificial intelligence and education. NJ: Ablex.
- Silva, A. A. (1994). Simulating electrical circuits with electronic spreadsheet. Computers and education. Vol. 22, n4, pp.345-53.
- Silver, E. A. (Eds.). (1985). Teaching and learning mathematical education. Reston, VA: National council of teachers of mathematics, Inc.
- Silver, E. A., & Thompson, A. G. (1984). Research perspectives on problem solving in elementary school mathematics. The elementary school journal, 84, pp. 529-545.



- Simon, H. A. (1990). Invariants of human behavior. Annual review of psychology, 41, pp. 1-20.
- Sleeman, D. H., & Brown, J. S. (Eds.). (1982). Intelligent tutoring systems. London: Academic Press.
- Sleeman, D. H., & Smith M. J. (1981). Modelling pupil's problem solving. Artificial Intelligence, 16, pp. 171-187.
- Sleeman, D. H. (1982). Assessing aspects of competence in basic algebra. In Sleeman, D., & Brown, J. S. (Eds.). Intelligent tutoring systems. New York: Academic Press.
- Sleeman, D. H. (1984). An Attempt to Understand Students' Understanding of Basic Algebra, Cognitive Science, 8, pp. 387-412.
- Sleeman, D. (1985). Basic algebra revisited: a study with 14-year-olds. International Journal of Man-machine Studies, 23, 127-149.
- Sleeman, D. H. (1987). PIXIE: A shell for developing intelligent tutoring systems. In Lawler, R. W. & Yazdani, M. (Eds). Artificial intelligence and education. Vol. 1. Norwood, NJ: Ablex.
- Sleeman, D. H., & Brown, J. S. (1982). Intelligent tutoring systems: An overview. In D. H. Sleeman, & J. S. Brown (Eds), Intelligent tutoring systems, pp. 1-11. London: Academic Press.
- Sleeman, D. H., Kelly, A. E., Martinak, R., Ward, R. D., & Moore, J. L. (1989). Studies on diagnosis and remediation with high school algebra students. Cognitive science 13, pp. 551-568.
- Softkey International Inc. (1994). The American heritage dictionary. (3rd ed.) (Ver. 3.5). Houghton Mifflin Co.

- Sternberg, R. J. (1985). Human abilities: An information-processing approach. New York: Freeman.
- Swan, M. B. (1983). Teaching decimal place value. A comparative study of conflict and positively-only approaches (Research Rep. No. 31). Nottingham, England: University of Nottingham, Sheel Center for Mathematical Education.
- Thompson, P. W. (1987). Mathematical microworlds and intelligent computer-assisted instruction. In G. Kearsley (Eds), Artificial intelligence and instruction, applications and methods. Addison-Wesley.
- VanLehn, K. (1982a). Empirical Studies of Procedural Flaws, Impasses, and Repairs in Procedural Skills. (ERIC Rep. No. LRDC-TR-ONR-8). School of Education, Virginia University.
- VanLehn, K. (1982b). Bugs are not enough: Empirical studies of bugs, impasses and repairs in procedural skills. The Journal of Mathematical behavior 3(2). pp. 3-71.
- VanLehn, K. (1990). Mind bugs: The origins of procedural misconceptions. London: MIT Press.
- Villano, M. (1992). Probabilistic student models: Bayesian belief networks and knowledge space theory. In Frasson, C., Gauthier, G., % McCalla, G. I., (Eds). Intelligent tutoring systems: Proceedings of Second international conference, ITS '92. Berlin Heidelberg: Springer-verlag.
- Viscuso, S. R., Anderson, J. A., & Spochr, K.T. (1989). Representing simple arithmetic in neural networks. In Tiberghien G. (Eds). Advances in Cognitive Science. Vol. 2. Theory and application. West Sussex: Ellis Horwood.

- Wagman, M. (1993). Cognitive psychology and artificial intelligence: Theory and research in cognitive science. Westport, Connecticut: Praeger.
- Wang, Y., & Garigliano, R. (1992). An intelligent language tutoring system for handling errors caused by transfer. In Frasson, C., Gauthier, G., & McCalla, G. I., (Eds). Intelligent tutoring systems: Proceedings of Second international conference, ITS '92. Berlin Heidelberg: Springer-verlag.
- Winograd, T. (1985). Frame representations and the declarative/procedural controversy. In R. J. Brachman & H. J. Levesque (Eds). Readings in knowledge representation. San Mateo, California: Morgan Kaufmann.
- Winston, P. H. (1992). Artificial intelligence, 3rd ed. Addison-Wesley.
- Woolf, B. P. (1987). Theoretical frontiers in building a machine tutor. In G. Kearsley (Eds), Artificial intelligence and instruction, applications and methods. Addison-Wesley.
- Yazdani, M. (1987). Intelligent tutoring systems: An overview. In Lawler, R. W. & Yazdani, M. (Eds). Artificial intelligence and education. Vol. 1. Norwood, NJ: Ablex.
- Young, R., & O'Shea, T. (1981). Errors in children's subtraction. Cognitive Science, 5, pp. 153-177.
- Zweng, J. J., Turner, J., & Geraghty, J. (1979). Children's strategies of solving verbal problems. Columbus, OH. (ERIC document reproduction service No. ED 178359).



## Appendix A

### *Mal-rule Collecting Tests*

The mal-rule collecting tests were used to collect students' mal-rules as well as their estimations on problem difficulty. Besides, problems in the tests were also used as materials for the comparing of the various measures on problem difficulty.

There were two tests: The first one being on simplifying of expressions composed of logarithms of numerical values and the second one being on both simplifying expressions of logarithms of variables and the solving of logarithmic equations. For each paper, there was a brief description of the paper and an instruction on how to finish it. There were also spaces for the students to fill in their personal information. As the tests were primarily designed to be used by Chinese students, the descriptions and instructions were both in Chinese characters.

---

---

## Mal-rule Collecting Test 1

---

---

### 香港中文大學教育心理系 數學測驗(一)

- 一) 本測驗的目的是找出同學計算對數(Logarithm)時可能遇到的問題，從而設計幫助同學的辦法。各題題目，少部份可能稍為艱深，但同學們只要不怕困難，盡自己的能力去做便可以了。
- 二) 答案可以寫在題目下面的空位上。如果需要草稿，亦請寫在題目旁邊，不要做在其他的紙張上。
- 三) 題目的旁邊，都有一個難度表上。請大家於計算每題後，評估一下該題的困難程度，然後圈上適當的數字。難度的分數由 1 至 5，1 為最容易而 5 為最難，餘類推。
- 四) 每位同學的計算過程、難度評分都有不同，也不一定有對或錯的分別。所以請大家不要與同學商討，嘗試自己作答。

請在下面的空位上寫上你的個人資料：

姓名： \_\_\_\_\_

班別： \_\_\_\_\_

學校： \_\_\_\_\_

性別： \_\_\_\_\_

## 測驗一 對數化簡

將下面的對數式化簡，並在旁邊的難度表上圈上題目的難度( 1 為最易，5 為最難，餘類推)。

Simplify the following expressions. On the difficulty scale beside each question, circle a number to indicate the difficulty level of the question. 1 is very easy, 5 is very difficult.

請將下列各式化簡為一數字 simplify the following expressions as real numbers:

已知 Given

$$\log 2 = 0.3010, \log 3 = 0.4771, \log 7 = 0.8451$$

1.  $\log 5 + \log 2$

1 2 3 4 5  
易 ☐ ☐ ☐ ☐ ☐ 難

(Ans = 1)

2.  $\frac{\log 16}{\log 64}$

1 2 3 4 5  
易 ☐ ☐ ☐ ☐ ☐ 難

(Ans = 0.6667)

3.  $\log 216$

1 2 3 4 5  
易 ☐ ☐ ☐ ☐ ☐ 難

(Ans = 2.3345)

4.  $\log 8 + \log 125$

1 2 3 4 5  
易 ☐ ☐ ☐ ☐ ☐ 難

(Ans = 3)



5.  $\log 60 - \log 6$   
 $\begin{array}{cccccc} & 1 & 2 & 3 & 4 & 5 \\ & \text{易} & & & & \text{難} \end{array}$

(Ans = 1)

6.  $\log 3 - \log 30$   
 $\begin{array}{cccccc} & 1 & 2 & 3 & 4 & 5 \\ & \text{易} & & & & \text{難} \end{array}$

(Ans = -1)

7.  $\log \frac{81}{2}$   $\begin{array}{cccccc} & 1 & 2 & 3 & 4 & 5 \\ & \text{易} & & & & \text{難} \end{array}$

(Ans = 1.6075)

8.  $\frac{\log 27}{\log 3}$   
 $\begin{array}{cccccc} & 1 & 2 & 3 & 4 & 5 \\ & \text{易} & & & & \text{難} \end{array}$

(Ans = 3)

9.  $\log \frac{100}{10} \times \frac{\log 100}{\log 10}$   
 $\begin{array}{cccccc} & 1 & 2 & 3 & 4 & 5 \\ & \text{易} & & & & \text{難} \end{array}$

(Ans = 2)

10.  $\log 7.5 + \log \frac{4}{3}$   
 $\begin{array}{cccccc} & 1 & 2 & 3 & 4 & 5 \\ & \text{易} & & & & \text{難} \end{array}$

(Ans = 1)

11.  $\log \frac{9}{4}$  1 2 3 4 5  
易 \_\_\_\_\_ 難

(Ans = 0.3522)

12.  $\log \frac{3}{5} - \log 12 + \log 2$   
1 2 3 4 5  
易 \_\_\_\_\_ 難

(Ans = -1)

13.  $\frac{1 + \log 2}{2 \log 2 + \log 5}$   
1 2 3 4 5  
易 \_\_\_\_\_ 難

(Ans = 1)

14.  $\log \left( \frac{10 \times 10}{10 + 10} \right) + \log 20$   
1 2 3 4 5  
易 \_\_\_\_\_ 難

(Ans = 2)

15.  $\log \frac{7}{9} + \log \frac{3}{28} - \log \frac{1}{2}$   
1 2 3 4 5  
易 \_\_\_\_\_ 難

(Ans = -0.7782)

16.  $\log \left( \frac{\log 10}{10} \right)$   
1 2 3 4 5  
易 \_\_\_\_\_ 難

(Ans = -1)

17.  $\frac{\log \sqrt{3}}{\log 243}$   
1 2 3 4 5  
易 難

(Ans = 0.1)

18.  $\frac{\log \sqrt{3} + \log 2}{2 \log 2 + \log 5}$   
1 2 3 4 5  
易 難

(Ans = 0.4147)

19.  $\frac{\log \frac{1}{4}}{\log(\log 100)}$   
1 2 3 4 5  
易 難

(Ans = -2)

20.  $\log 0.6$   
1 2 3 4 5  
易 難

(Ans = -0,2218)

\*\*\*\*\* END \*\*\*\*\*

- 請檢查是否已填上每題的難度
- Please make sure that you have circled the difficulty levels for each question.



---

---

## Mal-rule Collecting Test 2

---

---

### 香港中文大學教育心理系 數學測驗(二)

- 一) 本測驗的目的是找出同學計算對數 (Logarithm) 時可能遇到的問題，從而設計幫助同學的辦法。各題題目，少部份可能稍為艱深，但同學們只要不怕困難，盡自己的能力去做便可以了。
- 二) 答案可以寫在題目下面的空位上。如果需要草稿，亦請寫在題目旁邊，不要做在其他的紙張上。
- 三) 題目的旁邊，都有一個難度表上。請大家於計算每題後，評估一下該題的難度，然後圈上適當的數字。難度的分數由 1 至 5，1 為最容易而 5 為最難，餘類推。
- 四) 每位同學的計算過程、難度評分都有不同，也不一定有對或錯的分別。所以請大家不要與同學商討，嘗試自己作答。

請在下面的空位上寫上你的個人資料：

姓名： \_\_\_\_\_

班別： \_\_\_\_\_

學校： \_\_\_\_\_

性別： \_\_\_\_\_

本測驗分為兩部分，第一部分為對數式之化簡，第二部分為解對數方程式。兩部分均請作答。並請在旁邊的難度表上圈上題目的難度( 1 為最易， 5 為最難，餘類推)。

This test is divided into two parts, the simplification of logarithmic expressions and the solution of logarithmic equations. Please answer both parts. On the difficulty scale beside each question, circle a number to indicate the difficulty level of the question. 1 is very easy, 5 is very difficult.

\*\*\*\*\*

I) 請將下列對數式化簡

Simplify the following logarithmic expressions

1)

$$\log \frac{\sqrt{x}}{x}$$

易 1 2 3 4 5 難

(Ans = 0.5)

2)

$$\frac{\frac{1}{2} \log x}{\log \sqrt{x}}$$

易 1 2 3 4 5 難

(Ans = 1)

3)

$$\log x^2 - 2 \log x$$

易 1 2 3 4 5 難

(Ans = 0)

4)

$$\frac{\log x}{\log x - \log \sqrt{x}}$$

易 1 2 3 4 5 難

(Ans = 2)

5)

$$\frac{3\log\sqrt{x} + \frac{1}{2}\log x}{2\log\sqrt{x}}$$

 易 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 難

(Ans = 2)

6)

$$\frac{\log\sqrt{x} + \log x^3}{\log\sqrt{x} - \log x^4}$$

 易 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 難

(Ans = -1)

II) 解下列對數方程式

Solve the following logarithmic equations:

1)  $\log(x+6)+1=0$

 易 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 難

(Ans : x = -5.9)

3)  $\log(2x-5)=1$

 易 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 難

(Ans: x=2.5)

2)  $\log(9x-26)=2$

 易 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 難

(Ans: x=14)

4)  $\log(x^2+1)=1$

 易 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 難
(Ans : x =  $\pm 3$ )



$$5) \log \sqrt{x} = \frac{1}{3} \log x + 1$$

1 2 3 4 5  
易 難

(Ans : x = 1000000)

$$6) \log(x-3) - \log(x^2-9) + 1 = 0$$

1 2 3 4 5  
易 難

(Ans : x=7)

\*\*\*\*\* END \*\*\*\*\*

- 請檢查是否已填上每題的難度
- Please make sure that you have circled the difficulty levels for each question.

## Appendix B

### *Test on Solving Algebraic Equations*

The following shows the test paper used in studying tutoring strategies. There were three similar papers each with different numerical values but identical in structure. The papers were used as pretest, posttest and retention test respectively. In each paper, there was a brief description of the paper. As the tests were designed for the use of Chinese students, all descriptions were in Chinese. The following shows one of such papers used.

---

香港中文大學

教育心理學系

數學測驗(一)

- 本測驗目的是找出同學們的計算錯誤，從而設計幫助同學們的辦法。
- 本測驗需時約三十分鐘，同學們應有充份時間作答。
- 請盡量詳細作答，詳細列出 每一步驟，如需草稿，請寫在每一題右邊的空位上。
- 請將答案寫在答案簿上，勿弄污此問題紙。
- 在答題簿上，請寫上你的姓名，班別，編號。
- 本實驗的結果對全港同學均有很大的幫助，請盡量合作，努力作答。

謝謝你的幫忙。

## 解方程式

1.  $7X = 14$
2.  $4X = -12$
3.  $5X = 7$
4.  $8X = 18$
5.  $3X = 8 + 4$
6.  $3X = 4 \times 4$
7.  $3X + 4X = 14$
8.  $3X + 5 = 26$
9.  $7 + 4X = 19$
10.  $5X = 4X + 8$
11.  $3X = 3(2 + 3)$
12.  $12X = 2(3X + 3)$
13.  $7X = 2 + 4 \times 8$
14.  $17X = 19X + 25$
15.  $3 + 2X + 4X = 21$
16.  $4 + 3X + 4X = 25$
17.  $35X = 2 + 3(4X + 5)$
18.  $3 \times 2X + 3X = 19$
19.  $3(2X + 4) = 12(9 + 2X)$
20.  $21X = 3 \times 2(2X + 3)$



## Appendix C

### *Tutoring Scripts*

Three tutoring scripts were studied. In addition, each script was used in two cases: one with rehearsal and the other not. Hence there were totally six scripts.

The scripts reteaching and MBR were adapted from those used by Sleeman, Kelly, Martinak, Ward, & Moore (1989) and were translated into Chinese. The third script was designed in the present study and was originally in Chinese, it was translated into English for easy reference.

### **Tutoring Scripts: Reteaching without rehearsal:**

1. [FRESH PAPER]
2. Have student work the task aloud.
3. If wrong, say "THIS IS WRONG".
4. [FRESH PAPER]
5. say, " LET ME SHOW YOU HOW TO DO IT ..., THE REASON IS....( repeat the appropriate Manipulative rules) (Appendix D)"

### **Reteaching with rehearsal:**

1. Repeat steps 1 to 5 in above
2. GIVE PRACTICE TASKS

## **MBR without rehearsal:**

1. [FRESH PAPER]
2. Have student work the task aloud
3. After the student has completed the task, go back to EACH error, say:
4. " IT LOOKS LIKE YOU (DID) ....THIS IS WRONG BECAUSE... (Repeat the appropriate Manipulative rules)".
5. say," LET ME SHOW YOU HOW TO DO IT...., THE REASON IS ... (Repeat the appropriate Manipulative rules)".

## **MBR with rehearsal:**

1. Repeat steps 1 to 4 in above
2. GIVE PRACTICE TASKS

## **MBR with conceptual dissonance:**

1. [FRESH PAPER]
2. Have student work the task aloud
3. After the student has completed the task, go back to EACH error, say:
4. " IT LOOKS LIKE YOU (DID) ....THIS IS WRONG BECAUSE... (Repeat the appropriate Remedial rules)".
5. say," LET ME SHOW YOU HOW TO DO IT...., THE REASON IS ... (Repeat the appropriate Remedial rules)".

## **MBR with conceptual dissonance and rehearsal:**

1. Repeat steps 1 to 4 in above
2. GIVE PRACTICE TASKS

## Appendix D

### *Manipulative Rules Used In Solving Algebraic Equations*

The manipulative rules were originally from Sleeman, Kelly, Martinak, Ward, & Moore (1989). They were translated into Chinese when used.

1. Precedence -- Multiply or Divide before Adding or Subtracting (Mnemonic: "My Dear Aunt Sally")
2. "Get all the Xs to one side, all the numbers to the other"
3. "To undo added things, you subtract, to undo multiplied things, you divide."
4. "Whatever you do to one side, you must do the same thing to the other side."

The following shows the Chinese translation:

1. 先乘除後加減
2. 將有 x 的項移往左邊,將其他項移往右邊
3. 將一邊的數移往另一邊時,加的要變成減,減的要變成加.乘變除,除變乘.
4. 在方程式的一邊進行一個運作後,在另一邊一定要做相同的運作.
5. 當一數 a 乘兩數 b, c 的和時, a 應該分別乘 b, c, 然後相加, 即是說: $a*(b+c)=a*b+a*c$ .



## Appendix E

### *Remediation Rules Used In Solving Algebraic Equations*

These rules were used in the tutoring script MBR with CD. Originally, it was in Chinese but was translated into English here for easy reference.

#### Chinese Version:

- 依你看:  $(1+2)*3$ , 跟  $1+(2*3)$  有什麼不同? 如二者不同的話,  $1+2*3$  應取那一個值呢?  
請留意這式跟上述三者唯一不同的地方, 是有沒有括號及括號的位置. (等候學生回答).  
習慣上, 我們是先乘除後加減的. 即是說  $1+2*3 = 1 + (2*3)$ . 因為我們的答案的形式是  $x = \underline{\hspace{2cm}}$ . 如果不將所有沒  $x$  的項移往右邊, 而左邊只保留有  $x$  的項, 我們怎能得到答案呢?
- 將  $+3$  移往右邊, 要變成  $-3$  ( $-3$  變成  $+3$ ,  $*3$  變  $/3$ ,  $/3$  變  $*3$ ), 如果不這樣做的話, 我們看看會怎樣:  
 $x + 3 = 5$  變成  $x = 5 + 3$   
前式中的  $x$  只要是 2 便可以了, 但後式中的  $x$  則變成 8, 所以這是錯誤的.
- 例如說:  $1+2=3$ , 這是對的, 但如果我們只在左邊加 4, 即是說:  
 $3$ , 這對不對呢?  
(可改爲<sup>o</sup>  $1+2-4 = 3$ ,  $1+2 /4 = 3$ ,  $1+2 *4 = 3$ )
- 例如說:  $2*(3+4)$  應該是  $2*7 = 14$ , 而  $2*3+2*4=6+8=14$ , 所以  $2*(3+4)=2*3+2*4$ . 如果好像你剛才那樣做,....(模仿學生的做法, 但以數字代入)...., 左邊便不等於右邊.

#### English Version:

- What do you think the difference between  $(1+2)*3$  AND  $1+(2*3)$ ? If they are different, which one is equal to  $1+2*3$ ? Please note the only difference between the expressions is whether there are brackets and where the brackets are.

(Wait for answer)

We normally do multiplication or division before adding or subtracting.  
Hence,  $1+2*3 = 1 + (2*3)$ .

Because the answer should be in the form of  $x = \underline{\hspace{2cm}}$ .

If we do not move all terms without  $x$  to the right hand side, so that only terms with  $x$  at the left, how can we get the answer?

2. When you move  $+3$  to the right hand side, it should become  $-3$  ( $-3$  become  $+3$ ,  $*3$  become  $/3$ ,  $/3$  become  $*3$ ). If we do not do this, let us see what would happen:

$$x+3=5 \text{ becomes } x=5+3.$$

In the former  $x$  has to be 2 while in the latter,  $x$  should 8. Hence the above calculation is wrong.

3. For example:  $1+2=3$  is correct. If we add 4 to the left hand side only, then we have:

$$3$$

Is this correct?

(Use  $1+2-4=3$ ,  $1+2/4=3$ ,  $1+2*4=3$  where appropriate).

4. For example:  $2*(3+4)$  should become  $2*7 = 14$ , also  $2*3+2*4=6+8=14$ , hence  $2*(3+4)=2*3+2*4$ .

If I do it in your way,... (do as what the student did but with numbers to replace the variables)..., then left hand side cannot equal to right hand side.

Appendix F

List of Mal-rules

List of Mal-rules collected by using the mal-rule collecting tests. Only those with frequencies greater than or equal to 5 are placed here.

Code	Rule	freq.	Example
AA1	$\log(A \pm B) \rightarrow \log A \pm \log B$	64	$\log(2 + 3) \rightarrow \log 2 + \log 3$
AA2	$\log A \pm \log B \rightarrow \log(A \pm B)$	23	$\log 5 + \log 5 \rightarrow \log(5 + 5)$
AA5	$\log \text{Exp} = -\log A \rightarrow \text{Exp} = -A$	14	$\log x = -\log 10 \rightarrow x = -10$
AA6	$\log \text{Exp} = 0 \rightarrow \text{Exp} = 0$	5	$\log(x + 1) = 0 \rightarrow x + 1 = 0$
AA7	$\log(A \times B) \rightarrow A \times \log B$	23	$\log 2x \rightarrow 2 \times \log x$
AA8	$A \times \log B \rightarrow \log(A \times B)$	15	$2 \times \log 5 \rightarrow \log(2 \times 5)$
AA10	$\log A \rightarrow A \times \log;$	5	$\log 2 \rightarrow 2 \times \log$
	$\log A \rightarrow \log \times A$		$\log 2 \rightarrow \log \times 2$
AA12	$A \times \log B = \log C \rightarrow A \times B = C$	5	$2 \log x = \log 4 \rightarrow 2x = 4$
AB2	$\log(A + B) \rightarrow \log A \times \log B$ or	19	$\log(2 + 3) \rightarrow \log 2 \times \log 3$
	$\log(A - B) \rightarrow \frac{\log A}{\log B}$		$\log(x - 2) \rightarrow \frac{\log x}{\log 2}$
AB3	$\log(A \times B) \rightarrow \log(A + B)$ or	7	$\log(2 \times 3) \rightarrow \log(2 + 3)$



	$\log \frac{A}{B} \rightarrow \log(A - B)$		$\log \frac{9}{4} \rightarrow \log(9 - 4)$
AB4	$\log A \times \log B \rightarrow \log(A + B)$ or $\frac{\log A}{\log B} \rightarrow \log(A - B)$	7	$\log x \times \log x \rightarrow \log(x + x)$ $\frac{\log 2x}{\log x} \rightarrow \log(2x - x)$
AB5	$\log A + \log B \rightarrow \log A \times \log B$ or $\log A - \log B \rightarrow \frac{\log A}{\log B}$	31	$\log 5 + \log 2 \rightarrow \log 5 \times \log 2$ $\log 3 - \log 30 \rightarrow \frac{\log 3}{\log 30}$
AB6	$\log A \times \log B \rightarrow \log A + \log B$ or $\frac{\log A}{\log B} \rightarrow \log A - \log B$	23	No example found due to no chance $\frac{\log 16}{\log 64} \rightarrow \log 16 - \log 64$
AB7	$\text{Log. Exp.1} - \text{Log. Exp.2} \rightarrow \frac{\text{Log. Exp.}}{\text{Log. Exp.}}$	27	$\log x - \frac{1}{2} \log x = \frac{\log x}{\frac{1}{2} \log x}$
AB8	$\frac{\text{Log. Exp.1}}{\text{Log. Exp.2}} \rightarrow \text{Log. Exp.1} - \text{Log. Exp.2}$	34	$\frac{\log x}{\log x - \frac{1}{2} \log x} \rightarrow$ $\log x - (\log x - \frac{1}{2} \log x)$
AB9	$\text{Log. Exp.1} + \text{Log. Exp.2} \rightarrow \text{Log. Exp.1} \times \text{Log.}$	10	$\log \frac{7}{9} + \log \frac{3}{28} \rightarrow \log \frac{7}{9} \times \log \frac{3}{28}$ $\frac{1}{2} \log x + 3 \log x \rightarrow \log x (\frac{1}{2} \times 3)$
AB10	$\text{Log. Exp.1} \times \text{Log. Exp.2} \rightarrow \text{Log. Exp.1} + \text{Log.}$	10	$\frac{1}{2} \log x \rightarrow \frac{1}{2} + \log x$
AB11	$\log x \rightarrow x$ when not in an equation of the form $\log x = 0$	21	$-\log 10 \rightarrow -10$ $\log \frac{1}{10} \rightarrow 0.1$
AB15	$\log A \rightarrow \log(\text{value of } \log A)$	17	$\log \frac{1}{10} \rightarrow \log(-1)$

AB16	$\log A^n \rightarrow (\log A)^n$	5	$\log \sqrt{3} \rightarrow \sqrt{0.4771}$
AB19	$\log \frac{A}{B} \rightarrow \log A + \log B$	8	$\log \frac{3}{28} \rightarrow \log 3 + \log 28$
AB21	$\log A^n \rightarrow \frac{1}{n} \log A$	5	$\log x^2 \rightarrow \frac{1}{2} \log x$
AC1	$\log(A \times B) \rightarrow \log A \times \log B$ or $\log \frac{A}{B} \rightarrow \frac{\log A}{\log B}$	97	$\log(2 \times 3) \rightarrow \log 2 \times \log 3$ $\log \frac{2}{10} \rightarrow \frac{\log 2}{\log 10}$
AC2	$\log(A + B) \rightarrow \log A + B$ or $\log(A - B) \rightarrow \log A - B$	17	$\log(2 + 3) \rightarrow \log 2 + 3$ $\log(x - 3) \rightarrow \log x - 3$
AC4	$\log A + B \rightarrow \log(A + B)$ or $\log A - B \rightarrow \log(A - B)$	10	$\log(x + 6) + 1 \rightarrow \log(x + 7)$ $\log \frac{1}{3} - 1 \rightarrow \log(\frac{1}{3} - 1)$
AC5	$\log A \times \log B \rightarrow \log(A \times B)$ or $\frac{\log A}{\log B} \rightarrow \log \frac{A}{B}$	36	$\log 2 \times \log 5 \rightarrow \log(2 \times 5)$ $\frac{\log \frac{1}{4}}{\log 2} \rightarrow \log(\frac{1}{4} - 2)$
BA1	unable to reject roots that causes $\log(-ve)$	11	unable to reject the root 3 in $\log(x - 3) - \log(x^2 - 9) + 1 = 0$

---



---

## Appendix G

### *Teachers' Estimation of Problem Difficulty*

The following test was designed to collect teachers' estimation of problem difficulty. Problems taken from the mal-rule collecting tests are to be rated in a five-point scale. Also factors affecting the estimation were also to be rated. The questionnaire was in Chinese, English headings are added for better understanding only.



## 香港中文大學教育心理系

### 教師預測對數習題難度問卷

#### Instructions and Description of the Questionnaire

- 一) 在安排學生作業時，教師許多時候會根據學生的學業程度，選出適當難度的習題，以適應他們不同時期的練習需要。而在決定某習題的難度時，教師並不需要預先計算該題才知道該題是難或易。而只需要觀察題目，再根據個人經驗，便可以立刻作出決定。而此預測又經常甚為準確。本研究的目的便是嘗試找出教師在決定習題難度時的依據。再者，由於此預測的準確與否對學生的學習有一定的影響，因此本研究的另一目的便是要找出此預測的準確性。
- 二) 本問卷分為三部份：第一部份為一些簡單的個人資料，作用為找出這些個人因素是否對判斷題目難度產生影響。第二部份為嘗試找出影響教師在預測對數(Logarithms)習題難度時的因素。此處列出各種可能的因素，請各教師按其重要性評分。1 為最不重要，5為最重要，餘類推。如有未列出者，亦請於空位處加上。第三部分為習題難度評分：此處列出學生日常可能遇到的三十二題習題，請各教師按日常使用的方法，對每題的難度給予評分，然後於其旁的難度表上圈上適當的數字。難度的分數由1 至5，1為最容易而5為最難，餘類推。
- 三) 問卷填寫完畢後，請交回香港新界沙田中文大學教育學院李芳欒。謝謝。

請在下面的空位上寫上你的個人資料： (Personal Information)

性別：	男 ____	女 ____			
年齡：	20-25 ____	26-30 ____	31-35 ____	36-40 ____	>40 ____
教育程度：	中學 ____	專上學院 ____	大學 ____	大學及教育文憑 ____	碩士或以上 ____
數學為主修科目：	是 ____	不是 ____			
教學年資：	0-2 年 ____	3-4 年 ____	5-6 年 ____	7-8 年 ____	> 9 年 ____
任教中三，四，五級數學科年資：	0-2 年 ____	3-4 年 ____	5-6 年 ____	7-8 年 ____	> 9 年 ____

Difficulty Factors

下面列出各種可能影響教師在預測對數習題難度時的因素，請按其重要性圈上適當的數字，1為極不重要而5為極重要，餘類推。如有未列出者，亦請於表後加上。

預計解題過程中包括有特別容易犯錯的地方	極不重要	1	2	3	4	5	極重要
預計解題過程所需的步驟數目	極不重要	1	2	3	4	5	極重要
題目中所用數值的繁複程度	極不重要	1	2	3	4	5	極重要
題目中log出現的次數	極不重要	1	2	3	4	5	極重要
題目內包括的運作次數	極不重要	1	2	3	4	5	極重要
學生對該類題目的熟識程度	極不重要	1	2	3	4	5	極重要
其他（請列明） _____	極不重要	1	2	3	4	5	極重要
其他（請列明） _____	極不重要	1	2	3	4	5	極重要

下面為收集教師對題目難度的預測。在每題題目旁均有一難度表，請圈上適當的數字（1 為最易，5 為最難，餘類推）。

假設

$$\log 2 = 0.3010, \log 3 = 0.4771, \log 7 = 0.8451$$

$$11. \log \frac{9}{4}$$

易 1 2 3 4 5 難

$$12. \log \frac{3}{5} - \log 12 + \log 2$$

易 1 2 3 4 5 難

$$1. \log 5 + \log 2$$

易 1 2 3 4 5 難

$$13. \frac{1 + \log 2}{2 \log 2 + \log 5}$$

易 1 2 3 4 5 難

$$2. \frac{\log 16}{\log 64}$$

易 1 2 3 4 5 難

$$14. \log \left( \frac{10 \times 10}{10 + 10} \right) + \log 20$$

易 1 2 3 4 5 難

$$3. \log 216$$

易 1 2 3 4 5 難

$$15. \log \frac{7}{9} + \log \frac{3}{28} - \log \frac{1}{2}$$

易 1 2 3 4 5 難

$$4. \log 8 + \log 125$$

易 1 2 3 4 5 難

$$16. \log \left( \frac{\log 10}{10} \right)$$

易 1 2 3 4 5 難

$$5. \log 60 - \log 6$$

易 1 2 3 4 5 難

$$17. \frac{\log \sqrt{3}}{\log 243}$$

易 1 2 3 4 5 難

$$6. \log 3 - \log 30$$

易 1 2 3 4 5 難

$$7. \log \frac{81}{2}$$

易 1 2 3 4 5 難

$$18. \frac{\log \sqrt{3} + \log 2}{2 \log 2 + \log 5}$$

易 1 2 3 4 5 難

$$8. \frac{\log 27}{\log 3}$$

易 1 2 3 4 5 難

$$19. \frac{\log \frac{1}{4}}{\log(\log 100)}$$

易 1 2 3 4 5 難

$$9. \log \frac{100}{10} \times \frac{\log 100}{\log 10}$$

易 1 2 3 4 5 難

$$20. \log 0.6$$

易 1 2 3 4 5 難

$$10. \log 7.5 + \log \frac{4}{3}$$

易 1 2 3 4 5 難



21.  $\log \frac{\sqrt{x}}{x}$

易 

1	2	3	4	5
---	---	---	---	---

 難

22.  $\frac{\frac{1}{2} \log x}{\log \sqrt{x}}$

易 

1	2	3	4	5
---	---	---	---	---

 難 .

23.  $\log x^2 - 2 \log x$

易 

1	2	3	4	5
---	---	---	---	---

 難

24.  $\frac{\log x}{\log x - \log \sqrt{x}}$

易 

1	2	3	4	5
---	---	---	---	---

 難

25.  $\frac{3 \log \sqrt{x} + \frac{1}{2} \log x}{2 \log \sqrt{x}}$

易 

1	2	3	4	5
---	---	---	---	---

 難

26.  $\frac{\log \sqrt{x} + \log x^3}{\log \sqrt{x} - \log x^4}$

易 

1	2	3	4	5
---	---	---	---	---

 難 .

27.  $\log(x+6)+1=0$

易 

1	2	3	4	5
---	---	---	---	---

 難

28.  $\log(9x-26)=2$

易 

1	2	3	4	5
---	---	---	---	---

 難

29.  $\log(2x-5)=1$

易 

1	2	3	4	5
---	---	---	---	---

 難

30.  $\log(x^2+1)=1$

易 

1	2	3	4	5
---	---	---	---	---

 難

31.  $\log \sqrt{x} = \frac{1}{3} \log x + 1$

易 

1	2	3	4	5
---	---	---	---	---

 難 .

32.  $\log(x-3) - \log(x^2-9) + 1 = 0$

易 

1	2	3	4	5
---	---	---	---	---

 難

END

謝謝你的幫忙!

## Appendix H

### Learning Process Questionnaire

姓名: \_\_\_\_\_ 學校: \_\_\_\_\_  
班號: \_\_\_\_\_ 性別: \_\_\_\_\_

#### 學習過程問卷

#### 學習過程問卷是什麼

問卷內的問題是關於你對學校的觀感和你在學校中的學習方法。  
記著，學習並沒有所謂「正確」方法，而全在乎它是否適合你和你所讀的學科。  
如果你認為某一問題的答案，會因不同的學科而有所不同的話，請把問題應用於你覺得最重要的學科上，然後作答。

#### 如何回答

在每條問題下都有一行由 1 到 5 號的號碼：

1      2      3      4      5

請圈著你所選的號碼，以表示你的答案。

每個號碼代表以下的答案：

- 1 — 這句子對我來說，完全不適用或甚少適用
- 2 — 這句子對我來說，有時適用
- 3 — 這句子對我來說，大概一半時候適用
- 4 — 這句子對我來說，常常適用
- 5 — 這句子對我來說，永遠適用或差不多永遠適用

#### 舉例

「開著收音機做功課，我做得最好。」

如果你認為這句子對你來說是永遠適用，就請圈著 5 號如下圖：

1      2      3      4      5

如果你認為只是有些時候開著收音機對你做功課有幫助話，就請圈著 2 號，如下圖：

1      2      3      4      5

圈著你最先考慮的號碼空格——不要花太多時間思索。重要的是回答每一問題。

問卷內容絕對保密，請放心作答。謝謝合作。

## 學習過程問卷 B

請圈著你所選的號碼，以表示你的答案

每個號碼代表以下的答案：

- 1－這句子對我來說，完全不適用或甚少適用
- 2－這句子對我來說，有時適用
- 3－這句子對我來說，大概一半時候適用
- 4－這句子對我來說，常常適用
- 5－這句子對我來說，永遠適用或差不多永遠適用

1) 我只想讀那些將來能幫我找到工作的學科，而不是那些比較有趣的。

1      2      3      4      5

2) 我覺得學校的學習會給我一種愉快的感覺。

1      2      3      4      5

3) 我盡力在所有學科中取得高分，因為我想勝過其他同學。

1      2      3      4      5

4) 我只讀老師說要我們讀的東西，而不會再多讀。

1      2      3      4      5

5) 在學校學習中，我會去想想所學到的東西在現實生活中有多大用處。

1      2      3      4      5

6) 我有自己的方法去保存我的書本、筆記、和其他與上課有關的東西，使我容易找到。

1      2      3      4      5

7) 每當我測驗成績不好，便會擔心下次測驗的成績。

1      2      3      4      5

8) 我覺得應該把自己認為對的說出來，雖然其他人可能比我懂得更多。

1      2      3      4      5

9) 我很想在所有的學科上，勝過其他同學。

1      2      3      4      5

10) 牢記是我學習的最佳方法。

1      2      3      4      5



每個號碼代表以下的答案：

- 1 — 這句子對我來說，完全不適用或甚少適用
- 2 — 這句子對我來說，有時適用
- 3 — 這句子對我來說，大概一半時候適用
- 4 — 這句子對我來說，常常適用
- 5 — 這句子對我來說，永遠適用或差不多永遠適用

11) 當我閱讀新的資料時，我會聯想起學過的東西，其會對這些東西有不同的看法。

1      2      3      4      5

12) 我盡力訂定整學年的學習計劃，藉以使自己能夠得到最好的成績。

1      2      3      4      5

13) 我覺得我勤力讀書的唯一理由，是爲了將來離開學校後，可以找到一份理想的工作。

1      2      3      4      5

14) 只要我肯投入，很多學科都會變得很有趣。

1      2      3      4      5

15) 我喜歡測驗成績在班內張貼，因爲這樣能使其他人知道我比他們優勝了多少。

1      2      3      4      5

16) 我比較喜歡那些需要學習事實和細節的學科，多過那些需要理解的學科。

1      2      3      4      5

17) 我要在一個課題上有自己的意見，才會感到滿足。

1      2      3      4      5

18) 我一接到老師分派的作業，就盡快把它們完成。

1      2      3      4      5

19) 就算我在測驗前已用功溫習，仍會擔心自己可能考得不好。

1      2      3      4      5

20) 我發覺有些課題真能使我振奮。

1      2      3      4      5

21) 我寧願學業成績好多於受同學的歡迎。

1      2      3      4      5

每個號碼代表以下的答案：

- 1 — 這句子對我來說，完全不適用或甚少適用
- 2 — 這句子對我來說，有時適用
- 3 — 這句子對我來說，大概一半時候適用
- 4 — 這句子對我來說，常常適用
- 5 — 這句子對我來說，永遠適用或差不多永遠適用

22)在大部份學科中，我用功只求及格。

1      2      3      4      5

23)我盡力把在某學科中所學到的知識，跟其他學科中學到的聯繫起來。

1      2      3      4      5

24)我一下課就溫習，以確保自己能明白老師所授的內容。

1      2      3      4      5

25)老師不該期望我們去學習考試以外的東西。

1      2      3      4      5

26)有一天我或能改變世界上那些我現在認為是錯的東西。

1      2      3      4      5

27)任何學科，無論喜歡與否，我都會努力爭取最高分數。

1      2      3      4      5

28)對我來說記事實和細節，比要自己思考明白為佳。

1      2      3      4      5

29)我發覺學校教的東西，多數都很有趣，並會花額外的時間去加深我對它們的認識。

1      2      3      4      5

30)測驗卷派回後，我會改正所有錯誤，並嘗試了解犯錯的原因。

1      2      3      4      5

31)我只想留校讀書直至我有足夠條件找到一份好的工作。

1      2      3      4      5

32)我相信學校應幫助我對事物有自己的看法。

1      2      3      4      5

33)我視用功讀書為一項競賽，既參與其中，並要獲勝。

1      2      3      4      5

每個號碼代表以下的答案：

- 1 — 這句子對我來說，完全不適用或甚少適用
- 2 — 這句子對我來說，有時適用
- 3 — 這句子對我來說，大概一半時候適用
- 4 — 這句子對我來說，常常適用
- 5 — 這句子對我來說，永遠適用或差不多永遠適用

34)我不會花時間去讀那些我明知不會考的東西。

1      2      3      4      5

35)在課堂上討論過的課題，只要是有趣的，我都會用空餘時間去加深我對它們的認識。

1      2      3      4      5

36)老師認為應該要讀的東西，我盡力把它們讀完。

1      2      3      4      5

---完---

Biggs, 1992



Appendix I

Questionnaire on the Use of Electronic Homework

電子家課：學生使用意見調查表

姓名：\_\_\_\_\_ 班號：\_\_\_\_\_ 學校：\_\_\_\_\_

請用筆圈上你對下列各項問題的意見。5 表示你十分同意該句說話，1 表示你十分不同意，餘類推。

十分不同意

十分同意

<b>On Remediation 學習輔導方面:</b>					
I can have constant feedback in the working process 在我的學習過程中,電腦能於適當時間給予指導.	1	2	3	4	5
The feedback is helpful in correcting my errors. 電腦給我的指導能協助我改正錯誤.	1	2	3	4	5
The extra problems generated by the computer help me to consolidate the correct knowledge learned 每一題目算錯後,電腦產生的附加問題能加強我對正確算法的印象.	1	2	3	4	5
<b>On Arrangement of Problems 習題安排方面</b>					
The problems are arranged according to difficulty 習題能按難度安排	1	2	3	4	5
<b>About Input Method 輸入設計方面</b>					



I have no difficulty in using mouse as input device 對我來說,使用鼠標器作輸入工具,並無任何困難.	1	2	3	4	5
I have no difficulty in using keyboard as input device 對我來說,使用鍵盤作輸入工具,並無任何困難.	1	2	3	4	5
Although the input method in computer is different from writing on paper, it is easy to use 雖然電腦的輸入方法與我平日的習慣不同,但仍然十分容易使用.	1	2	3	4	5
I prefer using computer input method to do my homework 我比較喜歡用電腦輸入方法來做我的家課	1	2	3	4	5
<b>Sound and Image Effect 聲音,畫面方面</b>					
If an interesting picture or animation shows after a correct answer, I would be more motivated to learn 如果我答對了,電腦便顯示一幅有趣的圖畫或動畫,我將會更有 動機學習	1	2	3	4	5
If a piece of music follows a correct answer, I would be more motivated to learn. 如果我答對了,電腦便奏出美妙的音樂,我將會更有 動機學習.	1	2	3	4	5
<b>On Individual Student's Report 有關學生個人家課報告</b>					
The report helps me to understand my errors 此報告幫助我了解自己的錯誤.	1	2	3	4	5
The report helps me to know my result 此報告幫助我了解自己的成績.	1	2	3	4	5
The analysis on my performance given in this report is the information that I cannot have when doing homework by myself. 此報告中有關我的表現的資料,是我自己一人做家課時得不到的.	1	2	3	4	5
<b>On the Whole 整體來說</b>					
I can better master the learned materials by using Electronic Homework 使用電腦家課使我對所學的知識掌握得更好.	1	2	3	4	5
It is interesting 這套軟件甚為有趣	1	2	3	4	5
I can work at my own pace by using Electronic Homework 我能按自己的進度學習	1	2	3	4	5

The idea of Electronic Homework should be helpful in general 電腦家課應該是一個有用的工具	1	2	3	4	5
I hope I can do my homework on other topics on the computer 我希望能用電腦來做其他課題的家課	1	2	3	4	5
The computer was slow to respond to my input. 電腦對我的輸入反應太慢.	1	2	3	4	5
I would recommend it to my classmates. 我會推薦給其他同學使用.	1	2	3	4	5
<i>Personal Use of Computer</i> 個人電腦使用					
I frequently use computer. 我經常使用電腦.	1	2	3	4	5
If required, I can purchase a computer at home. 如需要的話,我可以購置一部家庭腦.	1	2	3	4	5

其他意見:

---

---



Appendix J

Teachers' Perception on Electronic Homework

電子家課：教師意見調查表

請用筆圈上你對下列各項問題的意見。5 表示你十分同意該句說話，1 表示你十分不同意，餘類推。

十分同意

十分不同意

On Remediation 學習輔導方面:					
Students can get constant feedback on progress 在學生的學習過程中,電腦能於適當時間給予指導.	1	2	3	4	5
The feedback is helpful in correcting students' errors. 電腦的指導能協助學生改正錯誤.	1	2	3	4	5
The extra problems generated by the computer help students to consolidate the correct knowledge learned 每一題目算錯後,電腦產生的附加問題能加強學生對正確算法的印象.	1	2	3	4	5
On Arrangement of Problems 習題安排方面					
The problems are arranged according to difficulty 習題能按難度安排	1	2	3	4	5
About Input Method 輸入設計方面					
There should be no difficulty for my students to use mouse as input device 對我的學生來說,使用鼠標器作輸入工具,並無任何困難.	1	2	3	4	5
There should be no difficulty for my students in using keyboard as input device 對我的學生來說,使用鍵盤作輸入工具,並無任何困難.	1	2	3	4	5



Although the input method is different from writing on paper, it is easy to use 雖然電腦的輸入法與學生平日的習慣不同, 仍然十分容易使用.	1	2	3	4	5
<b>Sound and Image Effect 聲音,畫面方面</b>					
If an interesting picture or animation shows after a correct answer, the students would be more motivated 如果我答對了,電腦便顯示一幅有趣的圖畫或動畫,學生將會更有動機學習	1	2	3	4	5
If a piece of music follows a correct answer, the students would be more motivated to learn. 如果學生答對了,電腦便奏出 美妙的音樂,他們將會更有 動機學習.	1	2	3	4	5
<b>On Individual Student's Report 有關學生個人家課報告</b>					
The report helps me to understand individual students' errors. 此報告幫助我了解個別學生的錯誤.	1	2	3	4	5
The report helps me to know individual student's result. 此報告幫助我了解個別學生的成績.	1	2	3	4	5
The analysis given in this report is the information that student cannot have when doing homework by himself or herself. 此報告中有關學生表現的資料,是他或她自己一人做家課時得不到的.	1	2	3	4	5
The report saves a lot of my time. 此報告節省了我許多時間.					
<b>On Student Group Report 有關學生組別家課報告</b>					
The report helps me to understand students' errors as a whole. 此報告幫助我了解整體學生的錯誤.	1	2	3	4	5
The report helps me to know results of the whole class of students. 此報告幫助我了解整體學生的成績.	1	2	3	4	5
The analysis saves a lot of my time in correcting students' exercise books. 此報告節省了許多我的改簿時間.	1	2	3	4	5
<b>On the Whole 整體來說</b>					
The computer was slow to respond to students input. 電腦對學生的輸入反應太慢.	1	2	3	4	5
Students can better master the learned materials by using Electronic Homework 使用電腦家課使學生對所學的知識掌握得更好.	1	2	3	4	5
It is interesting 這套軟件甚為有趣	1	2	3	4	5

Students can work at their own pace by using Electronic Homework 學生能按自己的進度學習	1	2	3	4	5
The idea of Electronic Homework should be helpful in general 電腦家課應該是一個有用的工具	1	2	3	4	5
I hope Electronic Homework can include other mathematics topics 我希望其他數學課題能包括在電腦家課內.	1	2	3	4	5
I would recommend it to other classes/ teachers 我會推薦給學生及其他教師使用.	1	2	3	4	5

其他意見:

---



## Appendix K

### *Students' Perception on the Use of Electronic Homework in Formative Evaluation*

Question	Mean
It is interesting 這套軟件甚為有趣	3.75
I can better master the learned materials by using Electronic Homework 使用電腦家課使我對所學的知識掌握得更好.	3.75
I can work at my own pace by using Electronic Homework 我能按自己的進度學習	4.5
I can have constant feedback in the working process 在我的學習過程中,電腦能於適當時間給予指導.	4.25
The feedback is helpful in correcting my errors. 電腦給我的指導能協助我改正錯誤.	4.25
The problems are arranged according to difficulty 習題能按難度安排	3.25
The report helps me to know my result 報告幫助我了解自己的成績.	3.75
The analysis of my errors I helpful 有關我表現的分析十分有用	3.5
The idea of Electronic Homework should be helpful in general 電腦家課應該是一個有用的工具	3.75
The analysis on my performance given in this report is the information that I cannot have when doing homework by myself. 此報告中有關我的表現的資料,是我自己一人做家課時得不到的.	4
I would recommend it to my classmates. 我會推薦給其他同學使用.	4.25

## Appendix L

*Results of Students' Perception on Electronic Homework*

<b>Question</b>	<b>Mean</b>
<b><i>On Remediation</i> 學習輔導方面:</b>	
I can have constant feedback in the working process 在我的學習過程中,電腦能於適當時間給予指導.	2.63
The feedback is helpful in correcting my errors. 電腦給我的指導能協助我改正錯誤.	2.78
The extra problems generated by the computer help me to consolidate the correct knowledge learned 每一題目算錯後,電腦產生的附加問題能加強我對正確算法的印象.	2.74
<b><i>On Arrangement of Problems</i> 習題安排方面</b>	
The problems are arranged according to difficulty 習題能按難度安排	3.06
<b><i>About Input Method</i> 輸入設計方面</b>	
I have no difficulty in using mouse as input device 對我來說,使用鼠標器作輸入工具,並無任何困難.	3.60
I have no difficulty in using keyboard as input device 對我來說,使用鍵盤作輸入工具,並無任何困難.	3.71
Although the input method in computer is different from writing on paper, it is easy to use 雖然電腦的輸入方法與我平日的習慣不同,但仍然十分容易使用.	2.91
I prefer using computer input method to do my homework 我比較喜歡用電腦輸入方法來做我的家課	2.38
<b><i>Sound and Image Effect</i> 聲音,畫面方面</b>	
If an interesting picture or animation shows after a correct answer, I would be more motivated to learn 如果我答對了,電腦便顯示一幅有趣的圖畫或動畫,我將會更有動機學習	3.22



If a piece of music follows a correct answer, I would be more motivated to learn. 如果我答對了,電腦便奏出美妙的音樂,我將會更有動機學習.	3.15
<b><i>On Individual Student's Report</i> 有關學生個人家課報告</b>	
The report helps me to understand my errors 此報告幫助我了解自己的錯誤.	2.94
The report helps me to know my result 此報告幫助我了解自己的成績.	2.82
The analysis on my performance given in this report is the information that I cannot have when doing homework by myself. 此報告中有關我的表現的資料,是我自己一人做家課時得不到的.	2.80
<b><i>On the Whole</i> 整體來說</b>	
I can better master the learned materials by using Electronic Homework 使用電腦家課使我對所學的知識掌握得更好.	2.75
It is interesting 這套軟件甚為有趣	2.55
I can work at my own pace by using Electronic Homework 我能按自己的進度學習	2.93
The idea of Electronic Homework should be helpful in general 電腦家課應該是一個有用的工具	2.96
I hope I can do my homework on other topics on the computer 我希望能用電腦來做其他課題的家課	2.91
The computer was slow to respond to my input. 電腦對我的輸入反應太慢.	3.88
I would recommend it to my classmates. 我會推薦給其他同學使用.	2.49
<b><i>Personal Use of Computer</i> 個人電腦使用</b>	
I frequently use computer. 我經常使用電腦.	3.07
If required, I can purchase a computer at home. 如需要的話,我可以購置一部家庭腦.	3.79



Appendix M

Students' Scores in Learning Process Questionnaire

Approach Scores					Deciles				
Schcode	Student No.	Sur	Deep	Ach	Deepach	Surdec	Deepdec	Achdec	Deepach dec
1	1	43	34	41	75	10	3	7	5
1	2	0	0	0	0	1	1	1	1
1	3	0	0	0	0	1	1	1	1
1	4	0	0	0	0	1	1	1	1
1	5	39	34	39	73	8	3	6	5
1	6	29	28	32	60	2	2	3	2
1	7	39	34	37	71	8	3	5	4
1	8	40	36	36	72	9	4	5	4
1	9	36	45	33	78	7	9	3	6
1	10	37	30	33	63	7	2	3	2
1	11	50	40	39	79	10	7	6	6
1	12	0	0	0	0	1	1	1	1
1	13	27	34	29	63	2	3	2	2
1	14	11	11	19	30	1	1	1	1
1	15	43	36	30	66	10	4	2	3
1	16	30	37	23	60	3	5	1	2
1	17	48	35	45	80	10	4	9	7
1	18	0	0	0	0	1	1	1	1
1	19	35	38	31	69	6	5	2	3
1	20	41	43	27	70	9	8	1	4
1	21	38	31	25	56	8	2	1	1
1	22	36	32	36	68	7	3	5	3
1	23	46	42	39	81	10	8	6	7
1	24	39	34	33	67	8	3	3	3
1	25	28	38	36	74	2	5	5	5
1	26	0	0	0	0	1	1	1	1
1	27	0	0	0	0	1	1	1	1
1	28	31	35	32	67	3	4	3	3
1	29	35	30	23	53	6	2	1	1
1	30	37	27	38	65	7	1	6	3
1	31	36	36	36	72	7	4	5	4

1	32	37	40	15	55	7	7	1	1
1	33	43	37	32	69	10	5	3	3
1	34	33	30	33	63	4	2	3	2
1	35	44	44	32	76	10	8	3	5
1	36	44	46	38	84	10	9	6	8
1	37	47	44	42	86	10	8	8	8
1	38	38	29	29	58	8	2	2	2
1	39	45	40	38	78	10	7	6	6
1	40	46	31	41	72	10	2	7	4
2	1	40	34	46	80	9	3	9	7
2	2	41	38	40	78	9	5	7	6
2	3	25	24	30	54	1	1	2	1
2	4	34	40	33	73	5	7	3	5
2	5	40	31	35	66	9	2	4	3
2	6	37	41	39	80	7	7	6	7
2	7	42	45	36	81	9	9	5	7
2	8	43	34	42	76	10	3	8	5
2	9	0	0	0	0	1	1	1	1
2	10	32	28	28	56	4	2	2	1
2	11	39	37	34	71	8	5	4	4
2	12	40	33	42	75	9	3	8	5
2	13	50	41	46	87	10	7	9	9
2	14	28	24	29	53	2	1	2	1
2	15	40	41	43	84	9	7	8	8
2	16	39	33	34	67	8	3	4	3
2	17	42	32	32	64	9	3	3	2
2	18	45	39	42	81	10	6	8	7
2	19	31	31	37	68	3	2	5	3
2	20	40	34	35	69	9	3	4	3
2	21	35	35	30	65	6	4	2	3
2	22	41	42	37	79	9	8	5	6
2	23	43	42	43	85	10	8	8	8
2	24	40	37	39	76	9	5	6	5
2	25	41	33	36	69	9	3	5	3
2	26	34	27	40	67	5	1	7	3
2	27	38	30	38	68	8	2	6	3
2	28	41	34	28	62	9	3	2	2
2	29	37	39	41	80	7	6	7	7
2	30	26	39	30	69	1	6	2	3
2	31	31	26	37	63	3	1	5	2
2	32	39	34	40	74	8	3	7	5
2	33	46	41	34	75	10	7	4	5
2	34	35	34	33	67	6	3	3	3
2	35	0	0	0	0	1	1	1	1

2	36	43	36	35	71	10	4	4	4
2	37	37	34	34	68	7	3	4	3
3	1	0	0	0	0	1	1	1	1
3	2	28	24	25	49	2	1	1	1
3	3	32	34	36	70	4	3	5	4
3	4	40	42	39	81	9	8	6	7
3	5	36	41	36	77	7	7	5	6
3	6	38	30	31	61	8	2	2	2
3	7	0	0	0	0	1	1	1	1
3	8	34	34	29	63	5	3	2	2
3	9	0	0	0	0	1	1	1	1
3	10	0	0	0	0	1	1	1	1
3	11	39	34	45	79	8	3	9	6
3	12	43	39	41	80	10	6	7	7
3	13	41	47	36	83	9	9	5	8
3	14	0	0	0	0	1	1	1	1
3	15	44	27	36	63	10	1	5	2
3	16	29	47	28	75	2	9	2	5
3	18	50	42	34	76	10	8	4	5
3	17	42	42	50	92	9	8	10	10
3	19	10	13	3	16	1	1	1	1
3	20	41	31	41	72	9	2	7	4
3	21	0	0	0	0	1	1	1	1
3	22	27	23	31	54	2	1	2	1
3	23	37	37	44	81	7	5	8	7
3	24	40	38	39	77	9	5	6	6
3	25	48	35	37	72	10	4	5	4
3	26	0	0	0	0	1	1	1	1
3	27	51	39	50	89	10	6	10	9
3	28	0	0	0	0	1	1	1	1
3	29	25	20	36	56	1	1	5	1
3	30	34	39	45	84	5	6	9	8
3	31	40	47	38	85	9	9	6	8
3	32	35	50	49	99	6	10	10	10
3	33	0	0	0	0	1	1	1	1
3	34	0	0	0	0	1	1	1	1
3	35	33	34	36	70	4	3	5	4
3	36	38	33	38	71	8	3	6	4
3	37	0	0	0	0	1	1	1	1
3	38	35	34	37	71	6	3	5	4
3	39	35	31	43	74	6	2	8	5
3	40	38	36	37	73	8	4	5	5
4	1	0	0	0	0	1	1	1	1
4	2	37	24	32	56	7	1	3	1



4	3	0	0	0	0	1	1	1	1
4	4	46	39	38	77	10	6	6	6
4	5	0	0	0	0	1	1	1	1
4	6	39	31	33	64	8	2	3	2
4	7	46	31	40	71	10	2	7	4
4	8	37	50	44	94	7	10	8	10
4	9	44	33	43	76	10	3	8	5
4	10	54	41	51	92	10	7	10	10
4	11	0	0	0	0	1	1	1	1
4	12	34	25	32	57	5	1	3	2
4	13	0	0	0	0	1	1	1	1
4	14	34	36	28	64	5	4	2	2
4	15	45	42	27	69	10	8	1	3
4	16	0	0	0	0	1	1	1	1
4	17	19	20	20	40	1	1	1	1
4	18	42	33	45	78	9	3	9	6
4	19	0	0	0	0	1	1	1	1
4	20	0	0	0	0	1	1	1	1
4	21	39	27	41	68	8	1	7	3
4	22	45	27	46	73	10	1	9	5
4	23	40	36	42	78	9	4	8	6
4	24	0	0	0	0	1	1	1	1
4	25	32	34	35	69	4	3	4	3
4	26	34	26	27	53	5	1	1	1
4	27	45	34	43	77	10	3	8	6
4	28	46	40	38	78	10	7	6	6
4	29	40	24	40	64	9	1	7	2
4	30	32	33	34	67	4	3	4	3
4	31	34	30	31	61	5	2	2	2
4	32	52	32	41	73	10	3	7	5
4	33	32	21	30	51	4	1	2	1
4	34	46	45	48	93	10	9	10	10
4	35	35	37	35	72	6	5	4	4
4	36	44	32	28	60	10	3	2	2
4	37	0	0	0	0	1	1	1	1
4	38	0	0	0	0	1	1	1	1
5	1	45	37	33	70	10	5	3	4
5	2	37	25	26	51	7	1	1	1
5	3	39	48	35	83	8	10	4	8
5	4	42	42	33	75	9	8	3	5
5	5	25	22	24	46	1	1	1	1
5	6	0	0	0	0	1	1	1	1
5	7	48	47	43	90	10	9	8	9
5	8	41	39	37	76	9	6	5	5

5	9	32	28	24	52	4	2	1	1
5	10	36	37	29	66	7	5	2	3
5	11	37	40	38	78	7	7	6	6
5	12	27	29	27	56	2	2	1	1
5	13	42	37	35	72	9	5	4	4
5	14	28	30	26	56	2	2	1	1
5	15	37	33	28	61	7	3	2	2
5	16	40	37	33	70	9	5	3	4
5	17	32	32	24	56	4	3	1	1
5	18	39	33	31	64	8	3	2	2
5	19	36	37	28	65	7	5	2	3
5	20	31	31	25	56	3	2	1	1
5	21	40	30	42	72	9	2	8	4
5	22	37	25	25	50	7	1	1	1
5	23	37	31	35	66	7	2	4	3
5	24	44	38	32	70	10	5	3	4
5	25	39	42	38	80	8	8	6	7
5	26	34	24	26	50	5	1	1	1
5	27	41	30	30	60	9	2	2	2
5	28	38	35	27	62	8	4	1	2
5	29	33	31	35	66	4	2	4	3
5	30	36	37	40	77	7	5	7	6
5	31	34	38	22	60	5	5	1	2
5	32	36	43	38	81	7	8	6	7
5	33	36	39	36	75	7	6	5	5
5	34	48	27	44	71	10	1	8	4
5	35	38	31	30	61	8	2	2	2
5	36	46	36	33	69	10	4	3	3
5	37	38	34	48	82	8	3	10	7
5	38	47	30	39	69	10	2	6	3
5	39	37	31	31	62	7	2	2	2
5	40	38	31	27	58	8	2	1	2
5	41	40	33	31	64	9	3	2	2
5	42	43	36	30	66	10	4	2	3
6	1	35	28	23	51	6	2	1	1
6	2	28	36	24	60	2	4	1	2
6	3	36	31	36	67	7	2	5	3
6	4	41	34	42	76	9	3	8	5
6	5	32	44	48	92	4	8	10	10
6	6	24	26	24	50	1	1	1	1
6	7	41	39	39	78	9	6	6	6
6	8	35	44	28	72	6	8	2	4
6	9	34	31	35	66	5	2	4	3
6	10	35	37	41	78	6	5	7	6

6	11	37	37	28	65	7	5	2	3
6	12	0	0	0	0	1	1	1	1
6	13	38	40	35	75	8	7	4	5
6	14	46	37	33	70	10	5	3	4
6	15	46	24	31	55	10	1	2	1
6	16	33	39	26	65	4	6	1	3
6	17	37	25	30	55	7	1	2	1
6	18	40	37	32	69	9	5	3	3
6	19	45	40	34	74	10	7	4	5
6	20	43	38	42	80	10	5	8	7
6	21	32	36	28	64	4	4	2	2

Note. Sur = Surface Approach; Deep = Deep Approach; Ach = Achievement Approach; Deepach = Deep Achievement Approach; Surdec = Surface Approach Decile Score; Deepdec = Deep Approach Decile Score; Achdec = Achievement Approach Decile Score; Deepachdec = Deep Achievement Approach Decile Score.



## Appendix N

### Homework 1

-----

香港中文大學教育心理系

數學練習(一)

對數化簡

- 一) 本練習的主題為對數 (Logarithm) 的化簡。同學們需要將本練習中的習題化至最簡。
- 二) 習題中需要的常數，已附於本練習中。同學亦可使用計算機，但要注意不要直接使用計算機算出答案。所有計算步驟必須詳細列出。
- 三) 答案可以寫在題目下面的空位上。如果需要草稿，亦請寫在題目旁邊，不要做在其他的紙張上。

請在寫上你的個人資料：

姓名：\_\_\_\_\_ 班號：\_\_\_\_\_

班別：\_\_\_\_\_

學校：\_\_\_\_\_

性別：\_\_\_\_\_

請利用下列的已知數字，將習題中各式化簡為一數字

simplify the following expressions as real numbers:

已知 Given  $\log 2 = 0.3010, \log 3 = 0.4771, \log 7 = 0.8451$

1.  $\log 0.6$

(Ans = -0.2218)

2.  $\log 216$

(Ans = 2.3345)

3.  $\log\left(\frac{\log 10}{10}\right)$

(Ans = -1)

4.  $\log \frac{81}{2}$

(Ans = 1.6075)

5.  $\frac{\log 16}{\log 64}$

(Ans = 0.6667)

6.  $\log 60 - \log 6$

(Ans = 1)

$$7. \quad \log 5 + \log 2$$

$$(\text{Ans} = 1)$$

$$10. \quad \log 3 - \log 30$$

$$(\text{Ans} = -1)$$

$$8. \quad \log \frac{9}{4}$$

$$(\text{Ans} = 0.3522)$$

$$11. \quad \log \frac{100}{10} \times \frac{\log 100}{\log 10}$$

$$(\text{Ans} = 2)$$

$$9. \quad \frac{\log 27}{\log 3}$$

$$(\text{Ans} = 3)$$

$$12. \quad \log \frac{15}{2} + \log \frac{4}{3}$$

$$(\text{Ans} = 1)$$



## Appendix O

### Homework 2

#### 香港中文大學教育心理系 數學練習(二)

#### 對數化簡及解對數方程式

- 一) 本練習的主題為對數 (Logarithm) 的化簡及解對數方程式。請同學們將本練習中的對數式化至最簡或是將對數方程式解出答案。
- 二) 習題中需要的常數，已附於本練習中。同學亦可使用計算機，但要注意不要直接使用計算機算出答案。所有計算步驟必須詳細列出。
- 三) 答案可以寫在題目下面的空位上。如果需要草稿，亦請寫在題目旁邊，不要做在其他的紙張上。

請在下面的空位上寫上你的個人資料：

姓名： \_\_\_\_\_

班別： \_\_\_\_\_

學校： \_\_\_\_\_

性別： \_\_\_\_\_

本測驗分爲兩部分，第一部分爲對數式之化簡，第二部分爲解對數方程式。兩部分均請作答。

(一) 請將下列對數式化簡

1)

$$\frac{\log \sqrt{3}}{\log 243}$$

(Ans = -1)

2)  $\log 8 + \log 125$

(Ans = 3)

3)  $\log \frac{\sqrt{x}}{x}$

(Ans =  $-0.5 \log x$ )

4)  $\frac{\log x}{\log x - \log \sqrt{x}}$

(Ans = 2)

5)  $\log x^2 - 2 \log x$

(Ans = 0)

第二部份 解方程式

6)  $\log(2x - 5) = 1$

(Ans:  $x = 7.5$ )

8)  $\log(x + 6) + 1 = 0$

(Ans:  $x = -5.9$ )

7)  $\log(9x - 26) = 2$

(Ans:  $x = 14$ )

9)  $\log(x^2 + 1) = 1$

(Ans:  $x = \pm 3$ )



## Appendix P

### *Students' Retention Test Scores*

School code	Student No.	Total score in part 1	Total score in part 2	Total score
1	1	11	1	12
1	2	2	0	2
1	2	2	0	2
1	4	-1	-1	-1
1	5	10	9	19
1	6	12	5	17
1	7	6	7	13
1	8	10	8	18
1	9	10	8	18
1	10	10	8	18
1	11	-1	-1	-1
1	12	-1	-1	-1
1	13	9	7	16
1	14	11	9	20
1	15	1	2	3
1	16	8	7	15
1	17	6	5	11
1	18	5	0	5
1	19	8	0	8
1	20	7	4	11
1	21	10	0	10
1	22	0	0	0
1	23	9	6	15
1	24	4	5	9
1	25	9	2	11
1	26	-1	-1	-1
1	27	-1	-1	-1
1	28	10	1	11
1	29	9	2	11
1	30	11	1	12
1	31	5	2	7
1	32	10	6	16
1	33	8	3	11
1	34	8	0	8
1	35	6	2	8
1	36	10	6	16

1	37	7	4	11
1	38	8	0	8
1	39	6	1	7
1	40	9	8	17
2	1	12	9	21
2	2	12	9	21
2	3	7	4	11
2	4	9	1	10
2	5	7	0	7
2	6	12	8	20
2	7	12	7	19
2	8	12	9	21
2	9	11	9	20
2	10	12	9	21
2	11	10	9	19
2	12	12	9	21
2	13	8	5	13
2	14	12	9	21
2	15	11	8	19
2	16	11	7	18
2	17	12	8	20
2	18	11	7	18
2	19	11	8	19
2	20	12	8	20
2	21	9	8	17
2	22	10	8	18
2	23	11	9	20
2	24	9	4	13
2	25	10	8	18
2	26	8	8	16
2	27	12	7	19
2	28	10	8	18
2	29	7	6	13
2	30	12	5	17
2	31	9	8	17
2	32	-1	-1	-1
2	33	12	9	21
2	34	12	9	21
2	35	9	6	15
2	36	12	5	17
2	37	12	9	21
3	1	-1	-1	-1
3	2	9	3	12
3	3	10	5	15
3	4	11	3	14
3	5	-1	-1	-1
3	6	12	5	17
3	7	12	9	21

3	8	12	4	16
3	9	-1	-1	-1
3	10	-1	-1	-1
3	11	12	5	17
3	12	11	5	16
3	13	12	9	21
3	14	-1	-1	-1
3	15	2	2	4
3	16	9	3	12
3	17	9	0	9
3	18	12	9	21
3	19	8	5	13
3	20	12	9	21
3	21	-1	-1	-1
3	22	9	6	15
3	23	11	5	16
3	24	-1	-1	-1
3	25	11	9	20
3	26	0	0	0
3	27	12	9	21
3	28	0	0	0
3	29	5	8	13
3	30	7	1	8
3	31	12	9	21
3	32	3	0	3
3	33	-1	-1	-1
3	34	-1	-1	-1
3	35	7	1	8
3	36	10	7	17
3	37	-1	-1	-1
3	38	4	1	5
3	39	-1	-1	-1
3	40	12	9	21
4	1	-1	-1	-1
4	2	12	9	21
4	3	3	0	3
4	4	11	3	14
4	5	9	3	12
4	6	12	9	21
4	7	7	1	8
4	8	9	8	17
4	9	8	6	14
4	10	11	8	19
4	11	12	9	21
4	12	3	1	4
4	13	7	7	14
4	14	4	4	8
4	15	6	9	15



4	16	5	1	6
4	17	10	3	13
4	18	12	9	21
4	19	5	0	5
4	20	-1	-1	-1
4	21	7	3	10
4	22	4	0	4
4	23	11	9	20
4	24	6	1	7
4	25	9	9	18
4	26	8	2	10
4	27	12	9	21
4	28	4	1	5
4	29	5	0	5
4	30	10	5	15
4	31	9	8	17
4	32	5	2	7
4	33	9	4	13
4	34	4	1	5
4	35	4	3	7
4	36	9	7	16
4	37	7	0	7
4	38	12	9	21
5	1	4	1	5
5	2	5	0	5
5	3	9	9	18
5	4	2	2	4
5	5	7	8	15
5	6	-1	-1	-1
5	7	4	0	4
5	8	-1	-1	-1
5	9	3	4	7
5	10	6	0	6
5	11	9	4	13
5	12	8	0	8
5	13	11	9	20
5	14	0	0	0
5	15	-1	-1	-1
5	16	0	0	0
5	17	10	9	19
5	18	9	9	18
5	19	8	0	8
5	20	-1	-1	-1
5	21	-1	-1	-1
5	22	5	6	11
5	23	3	2	5
5	24	7	2	9
5	25	5	3	8

5	26	3	1	4
5	27	5	6	11
5	28	6	2	8
5	29	8	3	11
5	30	1	0	1
5	31	12	9	21
5	32	6	1	7
5	33	8	2	10
5	34	7	3	10
5	35	1	0	1
5	36	5	1	6
5	37	9	5	14
5	38	-1	-1	-1
5	39	1	0	1
5	40	10	5	15
5	41	11	7	18
5	42	2	1	3
6	1	4	5	9
6	2	9	4	13
6	3	11	6	17
6	4	5	1	6
6	5	6	4	10
6	6	4	1	5
6	7	4	1	5
6	8	9	2	11
6	9	2	0	2
6	10	12	8	20
6	11	4	2	6
6	12	-1	-1	-1
6	13	7	6	13
6	14	6	0	6
6	15	11	3	14
6	16	10	2	12
6	17	12	2	14
6	18	7	3	10
6	19	7	2	9
6	20	5	1	6
6	21	4	4	8
6	22	6	0	6
6	23	10	5	15
6	24	8	2	10
6	25	6	4	10
6	26	6	0	6
6	27	4	0	4
6	28	7	1	8
6	29	7	3	10

---

Note. -1 = missing case

## Appendix Q

*Results of Teachers' Perception on Electronic Homework*

<i>Questions</i>	<b>Mean Score</b>
<b><i>On Remediation</i> 學習輔導方面:</b>	
Students can get constant feedback on progress 在學生的學習過程中,電腦能於適當時間給予指導.	3.5
The feedback is helpful in correcting students' errors. 電腦的指導能協助學生改正錯誤.	3
The extra problems generated by the computer help students to consolidate the correct knowledge learned 每一題目算錯後,電腦產生的附加問題能加強學生對正確算法的印象.	3.3
<b><i>On Arrangement of Problems</i> 習題安排方面</b>	
The problems are arranged according to difficulty 習題能按難度安排	3.5
<b><i>About Input Method</i> 輸入設計方面</b>	
There should be no difficulty for my students to use mouse as input device 對我的學生來說,使用鼠標器作輸入工具,並無任何困難.	4.2
There should be no difficulty for my students in using keyboard as input device 對我的學生來說,使用鍵盤作輸入工具,並無任何困難.	3.3
Although the input method is different from writing on paper, it is easy to use 雖然電腦的輸入法與學生平日的習慣不同,仍然十分容易使用.	3.2
<b><i>Sound and Image Effect</i> 聲音,畫面方面</b>	
If an interesting picture or animation shows after a correct answer, the students would be more motivated 如果我答對了,電腦便顯示一幅有趣的圖畫或動畫,學生將會更有動機學習	4



If a piece of music follows a correct answer, the students would be more motivated to learn. 如果學生答對了,電腦便奏出 美妙的音樂,他們將會更有 動機學習.	4.2
<b>On Individual Student's Report 有關學生個人家課報告</b>	
The report helps me to understand individual students' errors. 此報告幫助我了解個別學生的錯誤.	3.8
The report helps me to know individual student's result. 此報告幫助我了解個別學生的成績.	3.5
The analysis given in this report is the information that student cannot have when doing homework by himself or herself. 此報告中有關學生表現的資料,是他或她自己一人做家課時得不到的.	3.3
The report saves a lot of my time. 此報告節省了我許多時間.	2.7
<b>On Student Group Report 有關學生組別家課報告</b>	
The report helps me to understand students' errors as a whole. 此報告幫助我了解整體學生的錯誤.	3.7
The report helps me to know results of the whole class of students. 此報告幫助我了解整體學生的成績.	3.5
The analysis saves a lot of my time in correcting students' exercise books. 此報告節省了許多我的改簿時間.	3.8
<b>On the Whole 整體來說</b>	
The computer was slow to respond to students input. 電腦對學生的輸入反應太慢.	3.3
Students can better master the learned materials by using Electronic Homework 使用電腦家課使學生對所學的知識掌握得更好.	3
It is interesting 這套軟件甚為有趣	3.2
Students can work at their own pace by using Electronic Homework 學生能按自己的進度學習	3
The idea of Electronic Homework should be helpful in general 電腦家課應該是一個有用的工具	4
I hope Electronic Homework can include other mathematics topics 我希望其他數學課題能包括在電腦家課內.	3.8
I would recommend it to other classes/ teachers 我會推薦給學生及其他教師使用.	2.7

## Appendix R

### *Transcript of Students' Interview*

The following sections list the dialog between the researcher and seven secondary four students in Hong Kong. The students were chosen because there were systematic errors found in their scripts. The transcripts are in Chinese. However, translations of some of the lines can be found in Chapter 6.

=====

#### Protocol Analysis of S1

R:你就主要告訴我，當時你點解要咁樣計。

S:我想係唔記得點計！

R:但係你無論點樣一定有個方法,就算你唔記得,你當時唔記得啦,咁你由呢步到呢步,你都一定有個理由去做,或者你點解咁樣寫出來。

S:我寫了甚麼呀？

R:你log乘2加3呀,變了log乘0.301加0.4771

S:係呀！

R:點解你咁樣做法呢？

S:log乘2,唔係,唔係,2加3,咁樣做唔對嗎？

R:原因係乜野,點解咁做？

S:無呀,我見到個log2咁咪寫呢個,跟住log3就寫呢個,跟住加埋佢,唔係咁做咩？

R:咁個log呢！

S:係出面。

R:咁點解仲有log呢！

S:我諗唔記得點做啦！

R:唔得記點做,你因為log2就寫呢個0.301,log3就寫0.4771

S:係呀！

R:但係,log用去了,你個log仍然係番度。



S:可能我真係唔記得啦！

R:唔記得,唔緊要呀。

R:咁另外一條,你就寫成呢個啦,再寫成呢個,呢個對的,跟住你就約左 D2 去,係咪?點解咁做呢?

S:點解咁做呀?我想都有咯,咁下面,咁咪約左佢!

R:但係你知唔知道  $\log$ ,同埋普通一個,即係同無  $\log$  係完全唔同的,你知唔知道?

S:我諗我唔知呀!

R:唔知,即係你當左  $\log$  係一個乜野呀?

S:當左  $\log$  係咩?我諗我係無諗到。

R:即係當佢唔存在。

S:係呀!

R:當佢唔存在,咁  $\log$  實在約唔約得?

S: $\log$  約唔約得?

R:你宜家無左個  $\log$  喎,上面又。

S:約得掛!

R:都約得的。咁樣呀,咁跟住呢度就變成咁樣,呢度又點解 咁,呢度點解變成咁樣呢?

S:咁  $\log$ ,即係,2 乘,  $\log 2$  乘  $\log 2$ ,咁我咪將佢變左佢囉!

R:個  $\log$  仍然係番度。

S:係囉,都唔知點解果陣會咁樣做。

R:你宜家知唔知道做?

S:知唔知呀?我諗麻麻地掛,知 D 唔知 D 啦!

R:唔緊要,我地最緊要搞清楚當時點諗。

呢度就變成呢行啦,跟住呢度,係咪  $3 \log 2$  變成  $\log 0.301$  呢?

S:我諗係掛!

R:點解會變成,  $3 \log 2$  變成 0.301 呢?

S:我係無諗到個 3 喎,我無諗到前面果 D 架!

R:你睇漏左,頂係覺得佢唔緊要,唔需要理佢?果陣時你諗係點?

S:果陣時,我覺得,係,即係其實我係唔識做,都唔知點,即係唔記得左點做呀,咁我咪無理到個  $\log 3$  囉!

R:因為你唔知道點做,咁你就連個 3 都唔理,咁求其寫 D 野算數。



S:係呀！

R:你知道自己對不對呢？

S:我諗係錯架！

R:你諗住錯,呢度又係咁樣,求其寫D野落去。好啦,呢度,呢度又好似同一樣,係咪？

S:係啊！

R: $3 \log 2$ 加 $3 \log 5$ ,呢個就變成, $3 \log 2$ 就變成0.301, $3 \log 5$ 變成 $\log 5$ ,又係一樣？

S:係呀！

R:但係呢個0.301又變左0.903喎, $3 \log 2$ ,呢個點解？

S:我將個3乘左落去呀！

R:點解個3有時會出現,有時唔出現？

S:嘻嘻,唔知呀！

R:好啦, $\log 5$ 變成呢個2.0969點計架？

S:呢個呀,呢個唔記得左！

R:諗清楚。

S:呢個,係咪,將佢拆散跟住加呀又？

R:唔知你呀！

S:係囉,我都唔記得左呢個。

R:唔記得左。好啦,跟住落先啦。

S:跟住幾個我都答對喎！

R:有對的。

跟住呢 $\log 3$ 變成呢個3乘2加 $\log$ ,呢度係點樣做出來架？

S: $\log 3$ ,我將個 $\log 30$ 拆開晒呀！

R:30變成係.....

S:點解會咁？

R:係囉,點解？

S: $\log 3$ 加 $\log 2$ ？

R:會唔會係咁呢？我估咋, $\log 3$ 加 $\log 2$ 呢,實在即係 $\log 6$ ,呢個係 $\log 5$ ,係咪咁解？

S:噫,可能係喎,係呱！

R:即係話 $\log 30$ 變成 $\log 6$ 加 $\log 5$ ,好快喎你計得,跳左呢D步驟喎！

S:係咁呱,我諗。

R:跟住  $\log 5$  就變成  $\log 2$  加 3。

S:係呀。

R:好啦,跟住呢度呢,0.4771,呢度約左去喎,0.4771約左之後,仍然係度呢,點解?

S:睇下先,唔係,我果時可能睇呢,見到兩個  $\log 3$ ,跟住我咪寫左上去囉。

R:見到兩個  $\log 3$ ,咁呢度呀,一個  $\log 3$ ,是一個  $\log 3$  喎!

S:跟住,即係呢度仲有一個架嘛,跟住咪寫左落去囉,仲有嘛。

R:好啦,呢度呢第七題, $\log 81$  over 2, $\log 81$  除  $\log 2$ ,這個對不對?  $\log 81$  over 2 變成  $\log 81$  除  $\log 2$ ?

S:唔知呀,我諗當時係諗對的。

R:當成對的,跟住變成呢個,變成呢個, $\log 3$  over  $\log 2$  呀,呢度對的,主要係呢個步驟出錯。

好啦,呢度呢,睇睇呢個  $\log 10$ ,  $\log 100$  over  $\log 10$  變成  $\log 10$  係點解呀?

S:約左,我約左個零呀。

R:約左個零,約左個10,應該話,約左個10。呢度呢, $\log 4$  over 3 變成  $2 \log 2$  除  $\log 3$ ,又點解呀?

S:同呢個一樣,將佢拆左。

R: $\log 4$  除  $\log 3$ ,呢個係錯喎, $\log$  呢個 over 呢個唔等於  $\log$  呢個 over  $\log$  呢個架喎,記住呢條式啦!

跟住呢個,呢個 square root 點來的?  $\log 9$  over 4 變成 square root 3  $\log 3$  over  $2 \log 2$ 。

S:我都唔知點來的。

R:諗下,你試下,嘗試下。

S: $3 \log 2$ ,  $2 \log 2$ 。係咪果陣時我諗住約下,即係3乘3,2乘2咁呢?

R:即係9係等於3乘3,4係等於2乘2。

S:係呀!

R:咁所以9就變成,即係  $\log 9$  就變成  $3 \log 3$ ,4等於  $2 \log 2$  啦,但係你個 square root 點來架呢?

S:square root,咪係諗住將佢開左方咪無左囉,我諗係咁。

R:開左方就無左個3去。

S:係呀,當時我諗係咁。

R:你,即係,你想化簡佢。

S:係呀。

R:即係你想化簡左,所以就咁樣做。



跟住落到第十三題,  $1 + \log 2, 2 \log 2$  加  $\log 5, \log 5$  等於  $2 \times \log 3$ , 點解呢? 第十三題,  $\log 5$  等於  $2 \times \log 3$ 。

S: 呢度呀, 我諗住果陣時諗錯左掛!

R: 點樣錯法呢?

S: 諗住  $\log 3$ , 即係, 諗住  $\log 3$  加 2 呀其實可能係。

R: 即係 5 等於 3 加 2, 所以  $\log 3$ , 即係變左就係,  $\log 3$  加 2, 跟住 2 又出左去, 點解 2 又可以出左去架呢?

S: 就係諗錯左囉, 果陣時。

R: 點解你會咁錯呢?

S: 唔記得左點做。

R: 總之呢, 2 就可以隨時出入, 係咪咁解, 你印象中係咪咁解呢? 即係鍾意出去就出, 鍾意入番來就入番來。

S: 係記得 D 又唔記得 D 囉!

R: 即係記得有時得, 有時係唔得。

S: 係囉。

R: 但係你就無把握, 所以你就, 希望佢得就得啦, 係咪咁解呀?

唔緊要, 希望你下次改番佢啦, 呢個還掂唔計分。

呢度呢  $\log 5$ , 同呢個一樣啫, 你睇下, 係好奇怪啫, 呢度  $\log 5$  呢就變成  $2 \log 3$ , 呢度  $\log 5$  呢, 第十四題  $\log 5$  呢就變成  $\log 2$  加  $\log 3$  架啫, 係咪 D 有奇怪呢? 點解會有兩樣唔同的東西? 點解會咁架呢?

S: 就係, 即係, 記得唔係太清楚呀!

R: 記得唔清楚, 所以有有時就諗左一樣野, 有時就諗左另外一個做法, 係咪咁解呀?

S: 係呀, 係呀!

R: 咁呢度呢,  $\log 20$ , 就呢個  $2 \log 2, 2 \log 2$  加  $\log 3, 5 \times 4$ ?

S: 係呀。

R: 所以呢個又係  $\log 5$ , 所以呢個又係  $\log 5$ , 係咪? 呢個仍然都係  $\log 5$ , 不過呢就係又變成  $\log 2$  加  $\log 3$ 。

S: 係呀!

R: 呢個好奇怪啫,  $\log 5$  有時變呢個, 有時變呢個。其實, 因為兩個都唔肯定, 你覺得邊個才是對的?

S: 我覺得果個對的。

R: 呢個對。你係咪, 呢個兩次啫, 呢個一次啫?

S: 唔係。



R:好啦,跟住呢度呢,呢度好明顯第十六題啦,好明顯係  $\log 10 \text{ over } 10$  啦,係咪呀?變成  $1 \text{ over } 10$ ,呢個點解呢?

S: $\log 10$ 好似係等於1架,我印象中係。

R:等於1,係呀,你對呀。好啦,呢個對,跟住  $\log$  十分一等於1,又點解呢?

S:呢個唔知呀。我諗係無理到上面果個分子。

R:唔理個分子,變左  $\log 10$ ,你意思即係話呢個,呢個1就唔理佢,點解唔理呢?

S:因為唔知點計呀嘛!

R:唔知點計,所以就係唔理一D野,就做其他的,咁你相信對不對?

S:我相信係不對的,因果陣時都唔識做。

R:相信係不對。

好啦,2乘  $\log 3$ ,  $\log \text{ root } 3$  喎,係等於2乘  $\log 3$ ,呢個係有D問題架,係咪?

S:係。

R:點解呢,應該係咁樣架喎,或者點解你錯呢?

S:等於咩野我就唔記得啦,不過果陣時剛剛教緊,唔知咩野,3咩野  $\text{root}$  果D呢,咩等於3咩野  $1 \text{ over } 2$  果D,跟住唔知係咪咁咪做左上去囉。

R: $\text{root}$ 幾等於  $1 \text{ over } 2$ ,咁但係係2喎?

S:係呀,係呀,唔識做呀!

R:唔識做,所以就求其寫個落去。

S:係呀,我諗係。咁呀,唔做又唔得嘛,呢D野。

R:好啦,呢度呢,呀,呢度對的,呀,呢個D有問題,除,對呀。好啦,呢度呢,  $\log$  四分之一除  $\log 2$ ,變成爲  $\log 2 \text{ over } 4$ ,呢個主要問題點解要變成  $2 \text{ over } 4$  呢?

S:我轉左乘。

R:除,本來係除,你當左係乘,點解會當係乘呢?係睇錯左,淨係因為其他原因?

S:睇下先,我係變左做乘喎。

R:點解要變做乘呢?

S:因為見到可以變掛。

R:點解可以變呢?即係變左乘就簡單D,容易計D呀?即係你希望佢變做乘,咁呢就所變做乘。

S:係呀。

R:咁咪好奇怪,即係你希望佢咁樣,即係原來希望都會影響計數。

好啦,跟住呢就呢個  $2 \log 3$ ,呢度點樣呀,應該?呢度我都睇唔清實在你想點。

S:  $2 \log 3$  加  $\log 2$ ,  $2 \log 2$ , 噫, 點解咁奇怪?

R: 係囉, 點解呢?

S: 我寫左 D 左咩野呢? 係咪我掉轉左...呀, 唔係喎!

R: 呢個  $\log \sqrt{3}$  變成  $2 \log 3$  同上一樣啦。

S: 哎唷, 我諗呢度係寫錯左呀, 嘻嘻, 呢個係 3 字呀, 應該。

R:  $2 \log 3$ , 哦, 即係呢個唔要的。即係呢個呢就係, 3 變成 2 呢就因為係寫錯左, 咁跟住呢就係  $\log 2$  約左去, 約得的, 點解約得架?

S: 佢一樣呀嘛!

R: 見樣一樣就約得架啦, 係嘛, 咁加架喎, 你知唔知道?

S: 我當時諗住係架嘛, 唔緊要啦。

R: 好啦,  $\log 5$  變成呢兩個又點解?

S: 咪即係將  $\log 5$  拆開變  $\log 2$  加  $\log 3$  囉!

R:  $\log 2$  加  $\log 3$ ,  $\log 2$  加  $\log$  都係唔對的。

好啦, 跟住另外一張。二分一  $\log x$ , 你今次卻對了, 呢個係好特別的一點, 同埋上面頭先係一樣, 第幾題呀? 就係十七、十八同埋 paper one 的十七、十八, 係同一個情況, 不過你對了, 點解呢? 點解今次對, 上次錯呢?

S: 記得 D 囉, 即係唔同時間做架嘛兩份, 所以就記得 D。

R: 所以今次就記番, 記到。呢度呢, 呢度好奇怪, 你約左呢個, 跟住二分一除二分一係等於 0, 點解?

S: 唔係, 我係成個約架!

R: 成個約左去, 係成個約左去, 咁約之變成咩呢?

S: 1, 係咪變 1 架? 手快快囉, 我諗係, 諗住約左就無架啦咁囉!

R: 約左就無, 希望你...我地睇番第二題下, 就呢度呢  $\log x \text{ square}$ , 你今次呢又係同頭先一樣, 第三題, paper two, 點解今次又呢個 square 又變左做二分一呢?

S: 即係同果個諗法係一樣囉, 我諗。

R: 即係有時一個, 有時又另外一個。

S: 記得唔係咁清楚呀!

R: 記得唔係咁清楚, 所以有時係二分一, 有時就係 2, 即係你下一次又可能寫番呢個做乘  $2 \log x \text{ square}$ 。

S: 可能係。

R: 可能係咁樣。你今次呢預得對啦, 你同埋係隔離之喎, 同埋係隔離, 一個同一個真係, 差唔多相距都出現。

S: 但係都無乜時間睇番, 就咁做咁樣啦。



R:果樣對我來講唔重要啦,最緊要係睇下點解你,一時咁得意,一時呢個,一時果個。

咁,呢度又點解呢?  $\log x$  減二分一  $\log x$ ,點解係二分一?

S:係約左  $\log x$  呀!

R:約左  $\log x$ ,咁仲有減二分一架啲?

S:都唔知點解。唔記得左囉,一係就,應該。

R:你再諗多少少,會唔會有另外一D原因?

S:唔...我諗無啦。

R:無啦,唔緊要啦,我懷疑D原因係咪呢樣野,我一陣間先問你。

好啦,跟住呢度約左  $\log x$ ,變左  $3 \text{ over } 2$ ,呢個係  $-2$ ,跟住就變成爲  $-3 \text{ over } 7$ ,點解呢?

S:睇下先,2,哦,我諗呢應該係將2呢變左3,噫,點解會咁?

R:應該係呢個先對啲,可能你唔記得改番呢個。

S:可能係呀。

R:改番呢個,好啦,繼續。

好啦,跟住呢度,呢個好明顯係咁,因為你  $\log$ ,  $\log$ , 呢個第二 paper part two 第一題啦,  $\log x$  加6加1等於0,呢個就變成  $\log x$  加  $\log 6$ ,對唔對架呢?

S:呢個呀,我係乘左入去。

R:乘左入去,乘左入去呀?即係當  $\log$  係乘咁解,係咪咁解呀?

S:係呀。

R: $\log$  實在係咪乘呢?

S: $\log$  係咪乘呀!?

R: $\log$  同乘係咪一樣呢?

S:唔... $\log$  同乘係咪一樣呀,唔知呀!

R:好啦,跟住呢度又係一樣啦,係咪,呢度又係一樣啦,一樣啦。呢個又係一樣,你更加多,呢個  $x \text{ square}$  同二分一好有關係,係咪咁解?

S:係呀,撈到亂晒啦。

R:係啦,有D野唔係好清楚。呢個又一樣啦!呢度又做乜野呢?

R:呢度有D問題,由一又三分二  $\log x$  等於1,就轉成  $\log x$  等如1減一又三分二,呢個第五題,點解呢?

S:轉,轉左過去。

R:轉左過去,但係呢度係乘來架啲,呢度變成減啲!

S:我諗無留意呀。



R:總之轉過去就減。

咁呢度呢,係同頭先一樣啦,一樣啦,所以呢度唔駛再問你啦。

跟住就係,1減4,1減2  $\log x$ ,喂,呢個  $\log x$  唔見左啲,點解呢? 呢個第六題, $\log x$  等於0,跟住就變成減2等於0,點解唔見左呢?

S:漏寫呱,可能漏寫呀。

R:你係漏寫,頂係你係唔鍾意佢,所以唔要呀?

S:我諗係漏寫掛!

R:漏寫,好呀,咁無問題呀,咁我地完架啦。我想問番你一個問題,實在係頭先果度,不過我覺得有個奇怪地方,唔知你會唔會係咁樣。例如話呢度呢,我就唔知你係咪約錯啦,好似呢度一樣。例如話...我就有少少懷疑呢,你會唔會係將呢個當左係減呢,如果我當你呢個啦,二分一 over 二分一,你會唔會變成二分一減二分一咁樣睇?

S:我唔,我唔會。

R:唔會,不過就...咁呢個呢,呢個就漏左,唔係因為當左係  $\log x \log 2$  變左呢1減二分一就變成二分一,會唔會呢?

S:咁係約左  $\log x$ , 跟住...

R:約左  $\log x$  先,係嘛?  $\log x$  減二分一  $\log x$ ,你就約左  $\log x$ ,變左一減二分一。

S:係呀!

R:咁即係話呢,約左  $\log x$  啦, $\log x$  呢個係可以約的,你覺得。

S:我果陣時係咁諗。

R:我就最緊要知你時點諗,即係呢個減數約得。

S:我諗係見到相同掛。

R:好啦,咁我諗差唔多。

我無時間你講,本來我想同你講番點計,不過有人係出面等住啦。  
如果有需要的話,我再搵你。

S:好呀!

=====

#### Protocol Analysis of S2

R:我地宜家由 paper one 開始,就呢度呢第一題, $\log 5$  呢寫成  $\log 2$  乘3,呢個點解咁做呢?點解等於  $\log 2$  乘3呢?

S: $\log 2$  加  $\log 3$  呀嘛!

R: $\log 2$  加  $\log 3$ ,係,但係我宜家問題就係, $\log 5$  係等於  $\log$  乘3  
啲,呢度點解有咁情形出現?

S:我就係咁諗囉!

R:這個對啦,即係呢個變成呢個是對的!?

S:係囉,即係我初頭諗住果陣時亞sir講過話乘嘛,咁乘有陣時可以當係加架嘛!

R:乘可以變做加,所以...

S:係呀!

R:但係我想問妳呢就係5係咪等如2乘3呀?

S:唔係,應該係6。

R:咁點解可以變成咁呢?

S:咁...

R:因為妳想有呢個答案。

S:係呀,係呀!

R:咁,所以就將5夾硬變佢做2乘3。

好啦,跟住呢個對,呢度呢,呢個第三題,log 216變成log 2的3次方乘log 3的3次方,點解呢?

P:咁...我最主要係想係簡化佢囉,重,有通晒佢囉,例如呢個就2的3次方同埋log呢個3的3次方啦,咁樣整左出來之後呢,即係有相同個咩野囉,即最緊要佢呢度話log 3係呢個呀嘛,我最緊要係想take呢個number出來。

R:你最主要係想搵出log 2同log 3出來,所以好就係搵辦法變佢出來。

S:係呀。

R:但係如果由呢條式,有無D咩根據呢?呢條式你覺得對不對呢?

S:我做果陣時覺得對囉,我即係覺得...

R:覺得對的,記得有條式,咁樣呢條式即係點樣架?係咪即係話log 2乘3,2的3次方乘3的3次方,即係咁樣啦,3的3次方啦,係會變成爲log的2的3次方乘log的3次方,係咪咁呀?

S:係

R:意思即係話有條這樣的式:如果係log A乘B呢,就變成爲log A乘log B,有條咁樣的式?

R:我初初就係咁諗。

S:就係咁諗,唔緊要。

R:好啦,跟住我地去到第六題,log 30就變成log 3乘10,呢個對 啦,雖然你就無左個括號,但係我地唔理先啦。呢度,減log 3乘10你就寫成減log 3加log 10,原因係乜野呢?

S:我係分開兩個case咁睇囉,咁log 3咪咁樣log 3加log 10囉。

R:係,log 3加log 10.



S:即係同頭先果個一樣囉,同之前果...

R:咁但係個減號有無考慮?

S:咁,我初頭都以爲呢舊數減,同一個呢舊數囉。

R:呢舊數減呢舊數。

S:係,log 3呢,之前係講緊呢個,係講左比我聽係幾多架嘛,再加番log 10等如1呀嘛,咁咪呢個數減呢個再加呢個數囉。

R:你好多都對的,基本上。

跟住到第十二題,就log 12啦,係log 2的2次成乘3,乘3啦,就變成爲2乘log 2即係減2乘log 2再乘3,我諗你係漏左括號,係嘛?

P係。

R:應該有括號。

跟住呢,我主要睇呢個地方,減log 10 over 2變成爲減log 10減log 2減2log 2減3,乘3,跟住變成減2乘log 2,乘3來架?乘3一樣,無問題,即係照舊,但係再變成下面呢度呢,就變成左加,減2log 2加log 3,呢個問題係咪...點樣得出來呀?兩個都一樣。

S:由...由...即係...呢個log 2乘log 3嘛!

R:係呀,log 2乘log 3係等於...

S:即係我有少少make番之前果個錯處囉,以爲log 2乘log 3可以變成log 2加log 3囉!

R:係對的,呢個。

S:係囉,我係咁諗囉。

R:係對,但係你知唔知道宜家你錯係邊呀?

S:唔知。

R:係個減號問題呀!你宜家係減,我當呢個啦,減log 10 over 2,應該係減,拿個減號,因爲呢個log 10 over 2呢自己變成做log 10減log 2,咁如果再拆括號的時候,應該係減log 10加log 2,你係咪漏左轉符號呢樣野呢?

S:係呀。

R:但係你一向,一路都係咁做架喎。

S:唔係呀,因爲果陣時急得滯呀!

R:如果唔急就唔會咁架!?

S:係囉,因陣時亞sir催我地,嘩,咁多數重唔可以用計數機計喎,咁咪,做好快咁樣囉!

R:但係,我諗你計得晒架。

S:計得晒。

R:不過你急啲,係嗎?



S:我平時做數都係咁架啦,漏左D加符號錯,我地班好多都係咁。

R:平時都係咁錯,好多都係,你知唔知道呢?

S:但係後尾睇番就,即係,錯得好冤枉囉,係漏左,其實係對的,但係就正負號問題錯囉。

R:有無D咩理由令你D加減符號漏左呢?

S:有的,例如,好似,好似,例如,我覺得我地班呢好多有同一個錯處就係話,例如,好似,本來係 $-x$ 加 $y$ ,咁咪take左個負出來啦,就變成 $x$ 減 $y$ 啦,同一個錯處就係寫番 $-x$ ,即係負啦,take番出來啦,變成 $x$ 加番 $y$ 囉,唔記得左將佢轉符號囉,成日都係咁架啦!

R:唔記得左。

S:係呀,重有呢,有時計數呢,咁以為呢題咁淺啫,例如 $-5$ 加 $3$ 咁啦,咁就以爲咁淺就唔駛用計數機啦,咁左出來竟然漏左個負號,唔記得寫出來囉,咁咪錯好冤枉,成日都係咁。

R:係,好多都係咁,咁原因係咩呢?就係唔小心?

S:原因就係唔小心囉,重有以為覺得淺啦,好多時我地成班都係咁架啦,以為車條咁淺,都唔駛點樣,即係睇過一次之後呢,就無再睇啦,咁淺的話,都對架啦,唔駛再睇啦。

R:但係我覺得可能重有某一種原因,可能係習慣上咁樣做法呀!

S:我諗係有時,係囉有時係習慣上問題囉,重有有時緊張囉,最緊要緊張囉。

R:緊張就會咁架啦!

S:有時係架!

R:但係如果例如話我比,你假設個咁樣情況,如果好靜的時候,我比你係度慢慢做一條這樣的數,你會唔會錯呢個呢?

S:如果時間足夠的話,我諗絕對可以囉,即係,我諗我地班的話絕對可以架!

R:如果你錯左我地唔話比你聽,你會唔會都發現到呀?

S:都發現到架,有時會都發現到,即係如果係覆多幾次,有足夠時間,係啦!

R:若果唔覆,你就咁做呢,你仍然都會話錯?

S:係呀,係呀!

R:即係一種反應係咁樣錯。

好啦,跟住睇番第十三題呢度呢, $2 \log 2$ 加 $\log 5$ 你就寫成 $2$ 乘 $\log 10$ ,呢係點樣得出來?

S:咁,我之前呢都係,例如 $\log 10$ 的話呢,我就將佢成爲 $\log 2$ 乘 $\log 5$ , $\log 2$ 乘 $5$ 啦,咁其實...

R:即係呢個呢會變成2啦,log 2 乘 log 5,跟住呢就乘成2..

S:抽個log出來囉!

R:抽個log出來,2乘5,係咪呀?對唔對呢,你覺得?

S:我覺得應該對。

R:應該對的,有條咁樣式咩?有無架?

S:好似無。

R:哈哈,有無架?

S:好似無。

R:好似無,但係好似你覺得對的,實際上,你就無諗過有無呢條式,總之覺得就係咁樣。

S:我覺得係差唔多的形式囉。

R:係,差唔多係咁的形式。

呢度呢又係咁樣差唔多啦,log 1 over 4,呢個第十九題,log 1 over 4,你就寫成log 10減log 2的2次方,2的2次方是對的,log 10亦都對,log 10對唔對呢唔知啦,咁你話比我聽點解呀?

S:log 1 over 4係0.25呀嘛,咁我以爲,咁前面係,佢重緊要宜家化爲,化爲小數目,係,即係呢D,最緊要個答案係咁樣架啦。咁之前佢比過log 2我地架嘛,咁我就好似之前咁樣盡量將個number計出來係有關前面果D,果幾個囉!咁咪,諗囉。諗有D有乜野除左之後可以變成呢個數啦,0.25啦,諗下,諗下,呀,log 10減log 2的2次方,係喎,得喎,咪咁樣囉!

R:如果2我睇落去好似係log 1 減log 4先架喎,你無經過呢個步驟呀?

S:我無架,我直程...

R:你直程由呢個變出來,直接出來的,咁你又點知對唔對呢?

S:因爲我諗住...

跟住我自己覺得係,10的話,減呀嘛,減即係除啦,咁10減,10 over 4的話呢,即係都係差唔多啦都係,果陣時諗差唔多,都係。

R:你頭先講句話好得意架,減即係除下嘛,點樣叫做減即係除,即係,凡係減數都可以用除數來計。

S:我係咁囉。

R:你係咁諗,減即係除,除即係減。

S:除,未必架,我覺得...除係減囉。

R:下,點話?

S:除係減囉。

R:除係減,減係除。

S:有時,有時唔係咁諗架,即係睇情況,我都係有時。



R:幾時係咁諗呢？

S:我諗...宜家無 example 我又諗唔到呀就係。即係,我會睇做錯...

S:即係計緊數的時候,有陣時呢諗起就係話減即係除,無特別留心邊種情況係減,邊度係除。

R:係呀,我覺得差唔多係呢個,我就用呢個啦！

S:即係一種反應來,減即係除。

R:咁呢度未計完呀,得啦呢個。Part two,好得意架,D計錯的數。好啦,跟住呢個就係,呢度  $\log x$ ,呢個對的,呢個都對,拿,你有時對有時錯架D野。同樣一個情況呢,咁你有時會當...有時做對,有時做錯架喎！咁呢個呢就係減啦,抽左個  $\log x$  出來無錯啦,2分1減1,所以減到成負二分一減啦,係嘛？呢度跟住呢度就係,你呢個呢就係  $\log x$  乘負二分一,係咪？跟住  $\log x$  乘負二分一,點解呀？

S:咁我諗住, $x$  乘以呢個數呀嘛！咁我將  $x$  搬番上去,變成呢個咁囉！

R:唔,照乘落去,但係你實在有無留意到  $\log x$  自己係一個野來呢？跟住乘負二分一,係咁樣架喎！

S:有架！

R:有呀,但係都可以照乘得落去架！？

S:咁我初初以為呢,呢個負二分一都係有個  $\log$  係前面架囉。

R:都係係出面的,但係如果  $\log$ ,係咪即刻可以乘得入去架呢？

S:因為前面係抽左個  $\log$  出來架嘛,呢度。

R:呢度係抽左  $\log x$  出來喎,唔係抽左  $\log$  出來喎！？

S:我諗我睇錯左囉。

R:一樣啱？

S:係呀,我覺得係一樣。

R:因為  $\log x$  呢係可以抽得出來的,咁  $\log$  亦都可以抽到出來啱！

S:係囉,係囉。

R:呢度呢就係果個二分一減4對啦,呢個二分一減4變左二分一加四,點解呢？

S:我諗應該係符號錯。

R:點樣呢？

S:我諗係自己唔小心囉！

R:呢個係唔小心。

S:睇唔出自己冇咩錯。

R:應該無乜特別原因呵！係,無乜特別原因咁我地唔理佢啦。好啦,跟住到  $\log x$  加6,呢個第一題 section B 啦,你就寫成為  $\log x$  加  $\log 6$  加1等



如0,主要就係呢度。拿,呢度有兩個問題,第一就係由呢個步驟來到呢個步驟係點解?

S:  $\log x$  加  $\log 6$ , 即係我諗住係  $\log$  就係抽左出來啦, 咁應該係兩個同時可以有個  $\log$  架囉, 咁諗住  $\log x$  加  $\log 6$  囉。

R: 點解可以兩個同時都有  $\log$ ?

S: 因為我果陣時, 係諗住, 當抽左出來的話呢, 證明佢兩個同時都有  $\log$  架囉, 咁咪乘番落去囉!

R: 抽左出來就同時都有兩個  $\log$ , 咁然後呢,  $\log x$  加  $\log 6$ , 等如  $\log 6x$ , 呢個是對的, 是對的。不過你會唔會覺得好奇怪, 我唔知你會唔會留意到呢點架呢, 拿, 呢個  $\log x$  加 6 啦就變成  $\log x$  加  $\log 6$  啦, 又變番成  $\log 6x$  啦, 你唔知有無留意到  $\log x$  加 6 變左做  $\log 6x$  呢?

S: 無喎!

R: 無留意到, 好我地再睇呢度呀。呢度係同樣的情況喎, 又係點呀?  
 $\log 2x$  減, 變成  $\log 2x$  減  $\log 5$ 。

S: 係呀, 即係好似頭先咁囉, 以為  $\log$  係...(錄音帶盡要翻帶)

R: 繼續 paper two section B, 拿, 你係咪意思即係話如果係  $2x$  減就係變成  $2x$  減  $2y$  啦, 係咪咁呢? 係咪同呢個情況一樣呢?

S: 係呀, 係呀, 一樣。

R: 一樣, 但係有無留意到  $\log$  唔同呢?  $\log$  唔係呢樣野來架喎!

S: 無呀。

R: 無留意呢樣, 總之當左佢完全一樣。咁呢題數一樣啦, 呢個一樣啦。拿呢度對的, 呢段對, 呢度唔對。呢個又係一樣啦, 係嗎? 即係你相信佢一樣。

S: 即係有時候拎左運算果D來睇囉!

R: 當左佢乘咁樣睇, 有時候呀, 係咪經常性?

S: 咁又未必架!

R: 你知唔知道唔係架, 如果我...

S: 即係做果陣時呢係麻木左, 係唔知架, 根本就。

R: 哦, 做果陣時係麻木左, 即係, 即係點樣呀?

S: 做得多左以後呢, 就會覺得好似亂左咁架, 唔覺架啦!

R: 唔。呢度呢二分一  $\log x$  減三分一  $\log x$  等於1, 呢度對呀, 跟住呢就係  $\log x$ , 都對的, 都對的。呢度D有特別,  $\log x$ , 呢度係咩來架,  $\log x$  乘  $x$  over 6 等如  $\log 10$ , 係嘛? 呢度點解呢?

S: 睇下先, 抽出來啦, 我諗我漏左括號呀!

R: 漏左乜野呀?

S:漏左括號問題。

R:唔止漏左括號喎,唔止漏左括號喎。

S:仲有無端白事x擺上面。

R:無端端多左個x,係呀,點解呀?

S:睇唔到,唔知點解。

R:唔知點解,當係唔小心啦,若果係咁,一時、有時忽略左。呢個又係同樣問題啦,有兩個問題添喎呢度,你睇唔睇得出呀?不過一樣都係你過往的問題,第一個就係呢個log唔係乘啦,所以唔可以寫成咁,另外就係減呢就無變號啦。

S:應該係加。

R:即係你會經常犯呢兩個錯處。

S:經常犯,係我囉,係我自己經常犯呢個錯。

R:你自己都知道?

S:我知架!

R:不過知道都咁樣錯。

S:係囉,無辦法,成日都係咁架啦!

R:點解無辦法?

S:我覺得我自己做數呢成日都好粗心大意架,即係,漏係D好微的東西,係錯囉,成日都係咁。

R:呢個留心D就得架啦,應該。

係咪係出面等緊呀,你個同學?

S:唔知呢。

R:咁我諗都差唔多啦,就,主要你都係犯左果兩個錯處啲,其他D無咩大問題。咁好呀,唔該你。

---

### Protocol Analysis of S3

R:呢個就  $\log 64$ , paper one 第二題,就變  $8 \log 8$ ,點解呢?

S:因為  $8 \times 8$  等如 64。

R: $8 \times 8$  等如 64,但係呢個log來架喎, $\log 8 \times 8$  等如  $8 \times \log 8$ ?

S:我以為係囉,因為呢度又計得對,整數呀嘛,因為呢個係。

R:因為係整數。

好,呢度呢  $\log 216$  變成爲,呢個第三題,變成爲  $3, 3$  次方加  $2$  的  $3$  次方,點解呢?



S:因爲3的3次方係等如27。

R:27,係。

S:唔明點解？

R:27乘8。

S:係啦,應該係乘,乘8,係呀！

R:27乘8對呀,對呀,跟住呢出到來變成呢個啦喎！

S:因爲我諗將呢個3抽左出來,咁咪3加2咁樣囉。

R:咁意思即係話,3的3次方加2的3次方係可以變成爲3加2的3次方,係咪咁解？

S:係呀,因爲...

R:對唔對架呢個？

S:係呀,咁我抽出來,將指數又抽落前面咁咪抽落前面囉。

R:將指數抽落前面係對的,但係呢個步驟對唔對呢？

S:我估唔對掛。

R:嘻嘻,但係你果陣時覺得對的。

S:可能或者果陣時唔識做,咁咪我估唔對,但係都寫左落去先囉！

R:不過宜家識啦嘛！

S:應該識掛。

R:好啦, $\log 3$ 加2,跟住變成爲呢個加呢個,呢個有無問題呀,

你覺得？點解咁做？

S:因爲將 $\log 3$ 按計數機,我成日都按計數機架,按到呢個,再加番呢個,咪加呢個乘番呢個囉。

R:呢度係一條錯左的數式來架。

跟住呢,呢個係 $\log 60$ 減over  $\log 6$ ,有一條咁樣的公式,呢個點解咁做呢？

S:係減變左做除架嘛,好似話。

R:減變做除。

S:係呀,咁咪移左佢囉。

R: $\log 60$ 減 $\log 6$ 就變成爲 $\log 60$  over  $\log 6$ 。

S:係呀。

R:因爲就減變左做除,所有減都係變做除架啦。

S:好似唔係,但係又唔知點解寫左做咁樣囉。

R:幾時係減可以變做除呢,係果陣時諗。



S:陣時諗呀,果陣時諗住減呀,又見佢好似整數,咪變左做1囉。

R:即係,係咪因為有log的關係呢?即係見到log,就減就變做除。

S:係呀,因為睇番呢D,都係咁樣約左佢,咁咪約左佢囉。

R:呢D都係約左去,係喎,呢度對喎,因為呢D可以約左去,呢個就照約。  
但係呢個你約log 3架喎,呢個你約入面果個60同6喎。

S:咁我諗住,係,因為前面有log呀嘛,咁約左我估得掛。

R:咁個log約唔約得呀?

S:個log應該唔約得,咁,唔知,因為我見到呢度成個log 6 over 成個log 60  
咁樣囉。

R:係呀,即係60同6就約啦。

係呢度一樣啦,呢度一樣跟住啦。

呢度呢?

S:因為3, 3的4次方係81,咁就將番4調番呢度。

R:你,如果你做呢度的時候呢,你就變左做呢,log 81 over 番log 2先架喎,  
係咪咁解?

S:係呀!

R:點解可以咁做?

S:唔知呀,因為呢度咁log...我諗係其實唔係架,好似係唔係架。我記得  
就,咁但係果陣時咁樣做,唔知係又唔係好記得,咁咪寫番做log囉。

R:係,你當時唔係好肯定,呢個係最接近架勒。呢個最有可能啦。

S:係呀,因為我諗到3的4次方係等如81囉。

R:你係好快咁做出來,頂係諗一輪先做出來架?

S:呢個好似係諗諗地架,因為都猶疑緊點解難唔難呀。

R:呢個呢係7.5,就3乘10 over 2乘2呀。

S:呢個唔識做。

R:唔識做架。

S:係唔識呀,唔知點,淨係知道咁樣,又唔知點樣,點樣做好啦?跟住我  
都知係錯,但係唔知點樣做。

R:咁你點樣寫出來呀?你又唔知點做。

S:咁咪是旦寫下,睇下夾唔夾到囉,咁你log呀,咁你,應該係30除4,應該...

R:30除4,四、七廿八對呀!

S:係呀,應該等如7.5,咁呢度咪寫番2,但係然後就唔識解釋勒跟住。

R:跟住呢度呢第十一題,就log 3 square over 呢個2 square,跟住你就變成  
為2乘0.4771, 0.4771梗係log 3來架啦,咁呢兩行去邊架?呢兩行點計出

來？

S:將呢度,咁呢度前面指數計左啦。

R:係,即係變左咁樣先,係嗎？

S:係呀,咁跟住將指數搬左來前面,咁  $\log 3$  等於呢個,呢個等如呢個咪乘番佢囉。咁然後唔知點樣除㗎,除又唔知點樣變做減,唔知減抑或除。

R:除,除又點解變左做減呢又,好得意㗎！

S:我都唔知。好似因為果陣時話唔知  $\log$  幾等,唔知點樣變左做減或者係除,咁咪,其實唔係,應該都唔係,但係又唔知點解寫左做咁樣勒。

R:即係你宜家知道唔對勒,但係果時你就係咁樣寫。

S:係呀。

R:原因就係因為同  $\log$ ,即係有D同  $\log$  有關的東西。

S:係呀,唔知有時咩野除變左做減果D咁樣囉。

R:好得意呀呢樣野。

跟住呢度呢,呢個  $\log$ ,呢度第十五題啦, $\log 7 \text{ over } 9$  加  $\log 3 \text{ over } 28$  啦,前面果兩個,唔係唔係,跟住呢兩個變成  $\log 21 \text{ over } 28$  加  $3 \text{ over } 28$ ,  $3 \text{ over } 28$  係無  $\log$  的。

S:係咩?漏寫左㗎。

R:漏寫左㗎,係嘛?

S:唔知呀,應該係漏寫左啦,應該就。

R:但係呢個28有D問題㗎,九、三廿七㗎,抑或你計錯數?

S:應該我係用左來通分母,我見到。

R:係呀,用左來通分母,呢度只不過計錯數㗎,一時計錯㗎。

好勒,跟住呢,呢度呢就係等於10呀,二分一除五等於10。

S:應該係十分一先對呀,唔係㗎,十分一等如...

R:應該係十分一,係,點解等如10呢?

S:可能果陣時急得滯,咁咪寫左10。

R:好勒,跟住睇呢度, $\log 4 \text{ over } 1$  over  $\log 2$ .

S:咁我諗住,唔知幾多  $\log$  又約左佢㗎!

R: $\log$  約左去㗎, $\log$  約左去㗎,但係你頭先話比我聽  $\log$  唔約得架㗎,好似。

S:係呀,但係可能果陣時做得好亂,咁又求其,咁就約左佢,變左呢個數,咁又係整數,咁咪當係囉。

R:拿,你有時個  $\log$ ,就約得,有時個  $\log$  唔約得架㗎。



S:應該係唔約得架,我估。

R:應該唔約得,不過你果陣時就唔記得左呢樣野,就照做勒。

S:係呀,好急咁又咁整數,又變左做可以。

R:好勒,paper two.

paper two呢,就第一題,二分一,拿二分一  $\log x$ ,  $\log$  呢個  $x$ , 呢度勒,係咪第一個步驟變成爲呢個先?係咪呢個先?

S:係。

R:所以呢就錯左,呢度係錯的。

呢度呢  $\log x$  square,點解變成二分一  $\log x$  呢?

S:因爲,因爲...應該係2,但係寫二分一。

R:點解你會變成二分一呢?

S:或者可能當左係係,一時諗一諗,諗左以爲係呢個呀。

R:諗左過隔離?

S:係呀,唔係,即係諗左以爲係開方呀。

R:點解以爲開方呢?

S:係,但係唔知解,應該係2囉,但係唔知寫左開方喎!?

R:好,好得意總之呀,你錯得的。

第六題,呢度呢,就係二分一  $\log x$  呢個對啦下,加  $3 \log x$  又對呀,二分一  $\log x$  對呀,減  $4 \log x$  又對呀,好勒,跟住呢度就出左來勒。

S:跟住呢度係...係喎,我點走出來呢?

R:係,點計出來呢?

S:唔係,我未寫晒架,我寫左D,寫番落來呢度咋,我唔識計呀嘛!

R:哦,即係寫左D,呢度仲有D野。

S:呢度唔係答案來的,唔知呀,呢個唔係答案來,唔識做呀係。

R:咁得啦,好勒,呢度D你無做到。有時間同你地講番,叫先生同你地講番,有需要可以再搵我,唔該晒。

Protocol Analysis of S4

R:上次你做呢個測驗做過啦,今次就想睇番下呢,果D錯的地方,同埋想了解你點解錯。實在呢就你只要諗番下,我一路將張卷跟住睇下你做錯的地方,你話番比我聽你果陣時諗乜野,點解會咁樣做,唔需要理對頂錯,總之你話你覺得係咁樣做就得架勒,咁我就會錄音的,希望係唔介意啦。

咁宜家我地由 paper one 開始,呢個第一題啦,你見到就寫左  $\log 5$  寫成



log 3 加 2, 呢個對啦, 但係跟住你就寫成 log 乘 log 2, 呢度有無問題呢? 圈住呢度。

S:...

R: 點解咁做呢? ... 唔好意思, 我去門左度門先。

點解咁做呢?

S: 記得好似教過 D 咁野。

R: 有一條咁的式, 你記得。但係你記得, 即係咪好肯定條式係咁樣?

S: 唔係好肯定, 但大概有印象有條咁式, 所以呢就咁樣做法。實在果條式係加架, 你記唔記得? 宜家你可能知道, 應該係 log 3 加 log 2 架喎, 點解你記乘多過加呢? 你宜家當係乘啦, 實在應該係加架喎, 點解係乘多過加呢?

R: 想快 D 搵個答案出來, 加唔搵到個答案咩?

S: 即係將佢變左咁出來。

R: 咁但係你 log 3 加 log 2 都得架... 不過, 好似你剩係記得呢個, 總之你係記呢個, 唔記得果個!

好啦, 呢度呢, 第二題呢度係 log 2 的四次方 over log 2 的六次方, 咁你就變成爲 4 over 6, 咁就係約左, 係咪約左呢 D 野, 頂係點樣呢? 不過呢個對的, 係對的。

S: 係將呢個乘番咁多, 即係乘番四次方。

R: 乘番四次方, 呀, 當你對先啦。

好啦, 跟住呢度呢, 呢度未做完啲, 無問題。

呢度 3 乘 log 2 加呢個呢 log 125 呢, 呢個第四題啦, log 125 就寫成爲 log 3 加 2, 3 加 2 再乘 3 加 2, 即係 log 125 就寫成爲 log 3 加 2 點 3 加 2 再點 3 加 2, 呢個實在係點解呢?

S: 將佢拆左 3 個 5...

R: 3 個 5, 即係 3 加 2 係 5 來的, 即係 5 乘 5 乘 5。

S: 係。

R: 係, 但實在應該有括號係嗎? 你又諗唔起有括號!

S: 唔係, 當時係做完後邊, 因爲呢題諗唔到。

R: 哦, 即係唔諗住就寫低左先。

跟住呢度呢, 就係第六題先啦, 第六題呢度有個問題就係, log 30 就寫成 log 3 乘 2 乘 5, 呢個對架啦, 咁跟住寫爲減 log 3 乘 2 乘 5, 你就寫成減 log 3 加 log 2 加 log 5, 點解咁樣做呢?

S: 唔, 可能太急啦呢題。

R: 太急? 邊樣野錯左呀? 急到邊樣野錯左呀?

S:諗果陣時有D混亂囉！

R:唔係,我意思話,你話你果陣時因為急,令到邊樣野錯左呀？

S:正負。

R:正負,即係覺得應該有個負號的。拿,如果你呢個負號係對的,但係你的意思即話係減 $\log 3$ 加 $\log 2$ 啦,再加 $\log 5$ 咁解啦,你覺呢個係對的。但係有一個奇怪的地方喎,你頭先做呢題數的時, $\log 3$ 乘 $\log 2$ 呢,你就 $\log 3$ 乘 $\log 2$ , $\log 3$ 乘2你就寫成 $\log 3$ 乘 $\log 2$ ,呢度呢就係 $\log 3$ 乘2多一個乘5啲,你就變成 $\log 3$ 加 $\log 2$ ,拿,我地唔好理個5先啦,即係係呢個第六題呢,你就 $\log 3$ 乘2呢就變成 $\log 3$ 加 $\log 2$ ,係呢個第一題呢, $\log 3$ 乘2呢就係 $\log 3$ 乘 $\log 2$ ,兩個都係 $\log 3$ 乘2啲,點解一個係乘,一個係加？

S:唔知。

R:係咪好得意呀。總之你覺得係呢個時條呢個時候你就寫左呢個,係呢個時候你就寫左呢個。覺得呢個係對的,呢個係對的,係咪咁樣呀？

S:有陣時做呢D果陣時就會變左撈亂左！

R:咁,我意思係點解一時呢個,一時呢個呢？

S:...

R:唔知點解。係咪果時的反應係果個就寫果個呢？頂係有其他原因令你揀邊個呀？

S:果陣時的反應。

R:果陣時的反應,總之第一個出來果個你就寫果個,係咪咁解？

S:係。

R:拿,呢個就係 $\log 81$ ,第七題啦, $\log 81 \text{ over } 2$ ,81係4的3次方,我諗你對,呢個都對架。跟住呢度就係咁勒,4乘 $\log 3$ 減 $\log 2$ ,變成4乘0.4771減0.301,呢度括住呢個,你覺唔覺得...呢個係,你覺得呢個對唔對呀？

S:呢步唔對。

R:呢步唔對,點解呢,錯左係咩地方呀？

S:括番佢。

R:應該,應該括番佢。

S:括番個0.4771。

R:哦,應該呢度,果陣時點解要咁樣做法呢？

S:我諗係大意囉。

R:大意,會唔會經常都係咁樣架,頂係間中係咁架呢？

S:間中。

R:好啦,咁去第九題下。拿,呢度呢就 $\log 100 \text{ over } \log 10$ ,你就變成爲呢



個係10,點解咁計到出來?

S:除囉。

R:點樣除法?

S:將呢個10的2次方寫左10的10次方囉,跟住就10個 $\log 10$ 。

R:嘩,即係10的2次方變左10的10次方呀。果陣時係咁諗頂係你宜家係咁諗呀?

S:果陣時咁樣諗。

R:果陣時咁諗就變成呢個,點解10的2次方變成10的10次方呢?

S:10乘10囉我諗起。

R:即係10的2次方就諗起10乘10。

S:即係果陣時係諗起10乘10咁樣拆囉!

R:10的10次方,咁你就寫10的10...即係你諗住10乘10,於是就寫成10的10次方,係咪咁解呀?就唔係10的2次方。你意思即係話因為你諗起的係10乘10,10乘10,所以呢,跟住唔知點解就變左做10的10次方。你係諗到10的10次方,頂係無諗到呀?

S:有諗到10的10次方。

R:有諗到所以就變成呢個。

咁呢度未做完啲。跟住呢,呢度漏左少少野。呢度我諗你係...95...0.9542變左0.9541,呢個係唔小心啲,係咪咁解呀?你覺得係咪呀?0.9542,哦,是乘數。即係呢個2乘零點...呢個第十一題啦,2乘0.4771變成0.9541,呢個係錯係乜野地方呢?

S:2...

R:點解會漏左?你當然識計架啦,我諗。點解變成個1出來呢?

S:果陣時係將...一路咁樣乘落去。

R:係呀,一路咁乘呀,咁係漏左個2啲,係咪呀?乘漏左個2啲!

S:唔係好記得啦!

R:唔係好得,不過,應該你係識計呢D數,不過一時唔知點解啲!

好啦,跟住第十二題,呢個 $\log 12$ 呢就變成 $\log 3$ 減... $\log 12$ 就變成減 $\log 3$ 加 $\log 2$ 再加 $\log 2$ ,呢個點樣計出來架?

S:可以3乘2再乘2。

R:係啦,12係3乘2乘2。

S:跟住拆左佢變左咁。

R:係,就咁拆,即係好似頭先一樣,咁應該係減號減8呀嘛,係嗎?所以你就係錯左呢樣野。

S:係咪意思即係話當你有一時急,或者唔係好熟的時候呢,你就會錯呢



個。如果我問番清楚你,叫你問多次的時候,你可能做得到,如果你自己做就可能變左加。

R:咁第十三題, $\log 5$ 就變成爲呢一個,又點樣計出來架呢?

S: $\log$ ,跟住就3加2啦,變左 $\log 3$ 乘 $\log 2$ 。

R: $\log 3$ 加2就變成 $\log 3$ 乘 $\log 2$ 啲,有呢條式?

S:有。

R:有呢條式, $\log$ 括孤3加2係變成 $\log 3$ 乘 $\log 2$ ,有呢條式,有的!果條式係唔對架,不過我唔同你講住,果條式係掉轉左, $\log 3$ 乘2係等如 $\log 3$ 加 $\log 2$ ,但係 $\log 3$ 加2就唔係 $\log 3$ 乘 $\log 2$ 。跟住呢,我地睇第十五題呢,呢個 $\log 1 \text{ over } 2$ ,減 $\log 1 \text{ over } 2$ ,你就寫成爲減 $\log 1$ 減 $\log 2$ ,原因係咩呢?對唔對呢?你宜家覺得,錯係咩地方呢?

S:唔係上下都 $\log$ 有嘛!

R:咁你話比我聽點解你會寫呢個出來,即係減 $\log$ 二分一變成爲減 $\log 1$ 減 $\log 2$ ,原因係咩?

S:用除果條式變左 $\log 1$ 減 $\log 2$ 。

R: $\log 1$ 減 $\log 2$ 呀, $\log 1$ 減 $\log 2$ 呀,呢個減號應該點搞呀?

S:減號應該...

R:唔駛理佢,就咁擺係度。

S:括住佢。

R:括住,但係你唔括啲,點解呢?

S:好似前面D咁。

R:又係呢度咁,即係你都好經常犯呢個錯架播,雖然你知道係錯,但係你做的時候知唔知道錯呢?

S:唔知。

R:做的時候都係對的,做的時候都係對的。不過宜家,當我問番你的時候,我話比你聽錯時候,你就知道點解錯。但係如果唔話比你聽錯呢,你可能都覺得對的。

呢度呢就係 $\log 2$ 啦, $\log 2$ 對的, $\log 2$ 呢就變左做8乘 $\log$ 四分一,第十九題係點解呢?

S:四分一的八次方係2呀嘛。

R:四分一八次方係2?你真係咁諗?四分一的八次方係2?

S:係...八乘四分一係等於2。

R:你實在係邊個先?八乘四分一係等於2。

S:係。

R:因爲用呢個,呢個唔係,唔係呢個。

S:唔係。

R:你當時咁樣諗。好勒,咁宜家你個log呢,log 8乘四分一,咁係咪你本來係2變成8乘四分一啦,咁log 8乘四分一點解可以變成8乘log 四分一呢?

S:又係好似頭先log 100果個咁樣。

R:即係點樣?

S:即係變左...即係諗番起10乘10就變左10的10次方。

R:好呀。跟住呢log 5變成log 3減log 2啲,第十八題,點解呀?

S:我諗係做錯呢題。

R:點樣錯法?

S:應該係3加2。

R:3加2...都錯。即係第一,你就覺得應該log 3加log 2,不過就加變左做減,係咪咁解呀?

S:噫,會唔會係乘號來架?

R:可能係乘號,乘號都唔對啲,乘號都唔對啲!

S:錯左。

R:係咪你又記得有條咁的公式,當有條咁式,好啦,跟住呢度呢log 3乘2得啦,log 3加2乘2,呢度對架啲,無事呀!

跟住呢後面勒,呢度都對,對的,對的。呢度第五題,paper two section A 第五題,呢度呢3乘二分一log x加二分一log x,呢度變成爲3 log x,係點樣得到來?

S:唔..

R:3乘二分一log x加二分一log x係等如3 log x.

S:半個log x加半個log x變番一。

R:半個log x加半個log x等於一個log x,咁然後呢?

S:跟住變番一個囉。

R:咁即係一個log x啦,咁二分一個log x加二分一個log x就等如呢個啦。咁然後個3呢?

S:3乘番...

R:即係3乘番1個log x呢跟住就變成3 log x咁樣?

S:係。

R:有無先乘除後加減呢樣野呢?

S:!!...

R:唔記得左!



呢度呢,呢度對的。

$x^2$  等如 9,  $x$  等如 3, 對唔對? 有無留意呀?

S: 正負。

R: 正負, 果時又唔記得左啦。

呢度勒, 呢度第六題,  $\log x$  減 3 減  $\log x$  減  $3x$  加 3 呢, 就變成爲  $\log x$  加 3, 呢步驟係點樣計到出來? 睇唔睇到呀,  $\log x$  減 3 減  $3x$  加 3 變成  $\log x$  加 3。

S: 唔知呀!

R: 唔知道, 唔記得晒。盡量記一記, 會唔會係呢兩個有關係呢? 兩個約左。

S: 呢兩個減左。

R: 減左之後, 點解仲有  $\log$  番出來?

即係  $\log x$  減 3 就兩個減左去, 剩番一個  $x$  加 3 喎應該, 個  $\log$  係邊度來呢?

S: 唔... 諗唔起。

R: 唔知道, 好啦, 唔緊要啦, 有佢啦。

好勒, 我係呢度同你講完啦, 唔該你先!

=====

#### Protocol Analysis of S5

R: 宜家 paper one 第三題啦,  $\log 216$  變成  $3 \log 6$ , 跟住呢度無做, 呢個係點解?

S: 我無睇呢度囉!

R: 即係你唔知道 D 數字。

S: 係呀, 無睇到呀。

R: 跟住呢就到呢個第七題勒, 呢個都對的, 呢度有 D 問題, 你講我聽你點解會咁做?

S: 點解呀, 等我諗下先, 果陣時點解咁做, 唔記得啦!

R: 唔記得呀, 拿, 呢個  $4 \log 3$ , 你就當左佢分開, 左變左  $\log 3$  減  $\log 2$ ,  $\log 3$  減  $\log 2$  變成  $\log 3$  over 2 啦, 個 4 就係出面, 呢個做法對唔對架呢? 你自己宜家覺得?

即係例如話, 如果有一條數, 就係  $4x$  減  $y$ , 係咪可以變成  $4x$  減  $y$  架呢?

S: 唔得架!

R: 唔得, 應該變成爲乜野?

S:  $4x$  減  $4y$  囉。



R:  $4x$  減  $4y$ , 但係會唔會係呢個經常錯架呢? 你係同樣地錯。

S: 唔係, 呢 D...

R: 好啦, 既然唔記得就算啦。

咁跟住呢度呢, 就 7.5, 呢個  $\log 7.5$  呢, 宜家就變成 10 over 75, 點解呢?

S: 唔小心呢個, 係 75 over 10.

R: 肯定係唔小心, 無其他原因?

S: 係呀, 無呀。

R: 跟住, 好啦,  $\log 3$  減  $\log 2$  變成  $\log 1$ , 呢個係第十一題, paper one 第十一題,  $\log 3$  減  $\log 2$  成  $\log 1$ , 原因係乜野?

S: 唔小心囉。

R: 唔小心, 點解呢?

S:  $\log 3$  減  $\log 2$  係咩 3 over 2 呀嘛, 係嗎? 係呱! 我唔記得勒, 我做數好多時都係咁架啦。做數好多時都唔小心, 係咁啦。

R: 唔小心! ? 但係唔小心可能有 D 理由架喎, 就係因成日有 D, 有 D 好熟悉的觀念影響緊你, 所以, 都唔係話一 D 淨係話唔小心咁簡單的, 咁可能你會例如話, 突然間覺得, 呢個變成爲  $\log 3$  減 2 呀, 果個  $\log 3$  減  $\log 2$ , 好容易會變成  $\log 3$  減 2,  $\log 3$  減 2 變成  $\log 1$ 。即係, 我, 我會咁樣睇的話, 你就變成爲呢個, 3 減 2, 跟住變成  $\log 1$ 。拿, 你可能都話你係唔小心, 因爲你有時都會咁做, 如因果唔小心你會咁做, 所以會錯。你係咪咁樣呢, 如果你錯的時候, 原因係咪咁錯?

S: 3 舊野減 2 舊野等如 1 舊野囉, 如困睇番咁樣樣睇, 係囉!

R: 即係差唔多咁啱意思, 即係有時 D 野我地清醒的時候, 就係唔會錯的, 唔小心呢就會錯勒, 呢一題知道點做啦。

跟住呢度呢就係二分一  $\log 3$  加  $\log 2$ ,  $\log 4$  加  $\log 5$  呢度對的, 跟住變成, 你睇下呢個點樣。你個問題就咩呢, 二分一  $\log 3$  加 2, 呢個呢我諗就變成二分一  $\log 6$ , 跟住  $\log 20$ , 係勒, 二分  $\log 6$ , 對唔對先? 果陣時點解咁做呢?

S: 果陣時, 可能碰巧見番 D 數呢, 咁咪唔記得囉。

R: 唔記得左。但係呢樣野, 拿, 我頭先問你係話唔小心, 剛剛頭先係咪講到呢度呀, 係咪呢個呀? 呢個 4 呢, 你出左來, 係宜家呢個又係同樣情況架喎, 係咪? 一樣的東西喎, 咁所以呢, 有時就唔係話真係一時唔小心, 係, 有會唔會仲有一 D 咁樣野, 可能係一 D 習慣, 或者一 D 原因, 令到你會有 D 情況做到出來呢? 果個習慣係咩, 有咩原因呢?

S: 習慣? 唔知架。

R: 即係你頭先講果個, 呢個勒, 二舊野減兩舊野如一舊野, 呢個係一個你咁樣諗的東西, 咁但係呢度會唔會有類似咁樣的解釋?

第十八題, 第十八題, 會唔會呢?

S:唔記得嘍。

R:呢度有得佢唔知啦,跟住呢度呢,呢度加落來的,呢度點解我都唔知呀,點解會加多題?

S:你話加多題嘛。

R:唔緊要啦,跟住呢度呢就係,  $\log x$ ,  $\log x$  減二分一啦,呢度勒,呢度來到呢度,你可唔可以話比我聽點解咁做?好得意架,呢度都。

S:唔知呢。

R:唔知呀,我試一下。我試下話比你聽,你睇下係咪咁樣啦。拿,宜家  $\log x$  啦,呢個我諗你應該識計架,呢個都係  $\log$ , 即係變左二分一  $\log x$  啦,如果係咁樣的話,你就係  $\log x$  over 番個二分一  $\log x$ ,係咪?就變成咁樣勒,你諗下你果陣有無做到呢個,有無做到咁樣步驟?你諗清楚有無呢個步驟?

S:好似無喎。

R:如果無,咁出左先啦。拿,你個二分一  $\log x$  一定有地方黎,一係就咁樣,呢個減呢個變左二分一  $\log x$ ,唔要個分子。

S:有可能架呢個,呢個有能D。

R:有可能咁樣,就淨係因為個分子,即係呢個呢係由果個,由呢個分母得番來的,果個分子呢就係唔見左。

S:遊緊魂呀!

R:好牽強喎,個解釋。

仲有最後呢度勒,就  $\log x$  6次,六分一次方,二分一減三分一,六分一呀,就等於1。跟住呢個呢,  $x$  over 6 變成10,拿,呢個 比較特別,點解呢?即係呢行變做呢行,呢個係五題。 $\log x$ ,六分一等於  $\log 10$ ,  $x$  over 6 等於10。

S:我記得呢,就好似,唔知將呢度點樣整呢。就唔記左,呢個係唔肯定架啦,做呢題數果陣時。

R:你唔知對唔對的,係嗎?

S:係呀,所以就咁樣寫囉!

R:但係點解唔整呢舊野出來呢?

S:唔知呀。

R:會唔會係咁呀拿,我又估你下,你諗一諗佢。

好勒,第一變左下來,變左  $\log x$  over 6 啦,咁就等於  $\log 10$  啦,跟住最後可能係咁的問題,係咪咁呢?

S:唔係咁。

R:唔係咁。

S:不過係做完之後呢,都知道自己錯乜,後面即刻識點做啦。



R:好重要的線索黎假啲,對我來講。

S:可能作個答案出來都唔頂。

R:作個答案?

S:因為呢,答案作架嘛,有陣時。因為點呀,唔知佢呢,咁通常佢呢D數都係好整架嘛,無理由話佢會開方呀,會剩架嘛,可能作答案出來唔定。

R:即係希望個答案齊整D,就諗辦法做佢出來。

S:係勒,係呀。

R:可能係咁呀。

S:呢個其實呢,係咪約左個log去架,個log唔愛。

R:呢個可以約左log,  $x$  六分一次方等於10,  $x$  六分一次方同呢個數計。你應該識做播!

S:嘿,成日考試都係咁架啦。

R:下,你成日考試都係咁架?

S:我成日考試錯D唔小心果D野呢。

R:可能你又唔好夠熟,計得少D。

S:熟都無用,加減數都計錯。

R:咁又係,咁一係唔係好專心勒播。

S:睇做卷果陣時的心情啦。

R:有咩野會影響心情呀?

S:下,唔知架。果日心情好咪做好D囉,唔開心咪做差D囉,都會架嘛。同埋對住果份卷,有陣時見到佢咁煩呢,都懶得寫添呀!

R:但係呢,懶得寫都會架?

S:會架!

R:會唔會係去左諗第二D野呢,可能或者:

S:咁就少D,因為好多時都係被迫啦。係知道答案,但係知道唔計D分,咁就唔寫落去。

R:咁好啦,我都問到呢度勒,唔該晒你。

=====

#### Protocol Analysis of S6

R:咁宜家我地就由paper one開始,睇番下你有無做錯。拿,呢個64呢就寫成為 $8 \log 2$ 啦,呢度應該係2的6次方,唔係...咁你點解錯?

S:果陣時唔可以用計數機,計錯左。

R:計錯。好啦,跟住呢個第四題, $\log 125$ 你就寫成為 $2 \log 5$ ,點解呢?



S:唔知呀,可能係計錯數啦。

R:都係計錯數,呢度呢估你係125當左係5的2次方,唔知你係咪咁諗呢?都係計錯數呀,數字的錯處。好啦,跟住呢,呢度就...咁樣勒,去到第五,呢個呢減 $\log 2$ 乘3你就寫成減為 $\log 2$ 加 $\log 3$ ,你睇到原因係乜野呀?點解會咁做?

S:因為果陣時記錯左,就係呢個加呢個果陣時,唔係,呢個加呢個即係呢個乘呢個呀嘛,咁咪咁囉,調左下來。

R:唔,即係話, $\log 2$ 加2等於,2乘3就等於 $\log 2$ 加 $\log 3$ 。

S:係呀!

R:咁個減號就擺番落去勒!知唔知道要...

S:應該,呢度應該轉符號。

R:應該轉符號。你頭先同我一講一樣野呢就係話,咩呀,乘,乘呢就變成等於加呢個,你有無諗下點樣架,即係點樣乘變做加?

S:即係,因為有個定律就係 $\log$ 的 $m$ 乘 $n$ 係等如 $\log m + \log n$ 。

R:係,對,你係對的。咁跟住第六題呢,就係 $\log$ ,減 $\log 3$ 乘10啦,11啦,10,係,咁就減 $\log 3$ 加 $\log 10$ ,呢度同上面一樣勒啫?

拿,咁即係話有兩條數你都係,你宜家話比我聽你識,你應該轉符號,但係你果個時候唔轉架啫,點解會咁呢?

S:因為一寫落去唔覺,即係,我果陣時就知符號要轉囉!跟住宜家睇番知錯,咪知道要轉囉!

R:主要就係,點解你果陣時會錯呢?係果陣時你覺得反應就係咁樣架勒,頂係話你係寫低左呢個加,跟住再加個括號係上面,定點呀?點樣架?

S:即係,呢度乘咯,咁呢度咪加囉!咁係無...即係果陣時我就唔知負的話係要轉咯。普數就要轉,但係唔知 $\log$ 駛唔駛。

R:普通數就要轉, $\log$ 你就唔佑駛唔駛,所以你就唔轉。

S:係。

R:應該轉架。

跟住就係到呢個 $\log 75$  over 10,咁你就變成爲呢,呢個第十題啦,就 $\log 5$ 的3次方,呢個點解?

S:其實應該係75的3次,5的3次方係咪75呀?

R:5的3次方係125呀。

S:睇左5的3次方係75囉。

R:咁個10呢?啊,個10係下面啫,即係個數字錯。

跟住呢第十二題,我地見到有 $\log 2$ 呢,啊呢度減 $\log 12$ ,呢個減寫成呢個啦,減 $\log 2$ ,2的,2的2次方乘3,跟住呢就係減 $\log 2$ ,2的2次方即係 $2 \log 2$

啦,加  $\log 3$ ,原因就係...

S:同頭先一樣。

R:同頭先一樣啦。你經常咁樣錯架播!呢個呢,呢個又係一樣㗎,係嘛?

跟住呢就到  $\log 1 \text{ over } 2$ ,  $\log 1$  減  $\log 2$ ,你睇下呢個點樣錯。 $\log 10 \text{ over } 2$  寫成  $\log 1$  減  $\log 2$ ,呢個第十三題。

S:應該  $\log 10$  減  $\log 2$ 。

R:咁點解你果陣時寫  $\log 1$  呢?

S:我諗睇錯左,或者寫得快得滯。

R:寫得快得滯,係嗎?意思即係,你係無其他原因的,唔會話  $\log 10$  等於  $\log 1$  的,不過一時唔小心,係嗎?咁,拿,  $\log 1$  減  $\log 2$  呢,呢度又有D奇怪,  $2 \log 2$ ,  $\log 2$ ,跟住呢個  $\log 1$  你就寫成1,呢個原因係乜野?拿,我睇到  $2 \log 2$  加  $\log 1$  減  $\log 2$ ,你就變左一個  $\log 2$  啦,加番1,係嘛?  $\log 1$ ,加落去的。

S:我諗係睇到呢個係  $\log 10$  囉,跟住呢個...

R:即係呢個只係寫錯左,呢個又對番。

實在我搵你來唔等於你地低分架,千祈唔好誤會下,唔你低分,不過係有D野我想了解下㗎。

呢度呢,係二分一  $\log x$  減  $\log x$  啦,呢個未做完㗎。點解唔做落去㗎,其實。點解呢度唔做落去呢?

S:呢度唔計埋落去,唔知呀!

R:唔知點解,唔緊要。

好啦,跟住呢幾得意,  $1 \text{ over}$ ,呢度第四題,  $1 \text{ over } 1$  減二分一,你就寫成爲0.2啦,點樣得來?

S:呢度囉。

R:10除0.5呀,即係1除0.5,1除0.5,等於10除5㗎。

S:咁,我諗我係計錯左數。

R:D數字計錯左,你D數字係太好呀,計錯左係D數字果度。

跟住呢度呢,呢度第六題,section A,咁,我呀見到呢度對的,出到來係二分一加三乘  $\log x$ ,對的,二分一減4乘  $\log x$  都對的,咁你係約左  $\log x$  啦,好似下?

S:係呀!

R:跟住呢個係  $7 \text{ over } 2$  乘  $9 \text{ over } 2$  呀?

S:係呀!

R:點解呢?

S:因爲呢度,咁,有個,即係有個符號架嘛,因爲一個數,一個負數除一個



負號,係將佢上下顛倒呀嘛。

R:負數除一個負數上下顛倒?

S:唔係,係一個分數除一個分數上下顛倒。

R:係呀,係呀,係呀,咁呢個  $9, 2 \text{ over } 9$  對唔對?

S:加呀可?點解計到,減計左加。

R:減就,睇錯左加定係點呀?定係點呀?

S:我諗我又睇錯數嘍!

R:計錯左定係點呀?

S:係囉,應該寫快得滯,成日都睇錯。

R:你都,D數字都經常計錯數字?拿,計到呢度唔係容易架,好多人唔到呢度架,你根本計到,就呢度後尾就唔得。

呢度問題出現係咁勒, $\log x$ 減3呢, $x$ 減3係 $y$ 啦, $\log x$ 減3係 $y$ ,呢個對的。好啦,宜家問題呢就係話你果個 $\log x$  square,減添播,減9,你就變成爲加 $y$  square,咁你解釋一下點解咁樣呢?

S:唔知,無諗呀,寫得快得滯啫,跟住就成個數就計唔到落去啦!

R:係呀!係呀,跟住果D就全部錯晒,因爲你一開始已經錯左。

S:應該又係睇錯題目囉,即係睇果陣時,呢度“tun”,呢度..

R:點解會係 $y$  square呢,呢個?

S: $y$  square?點解?

R:呢個係 $y$ ,呢個係 $y$  square.

S:哦,哦,我知我睇錯咩啦,睇錯左呢個係 $9, 3$ 的square呀嘛,咁咪照抽個square出來囉,其實係應該有個 $6x$ 係中間,係囉。

R:哦,哦,但仲有一個,拿,咁即係話呢個你當左係 $3$  square,即係意思第一樣野錯啫,你當左 $x$  square減3的square呢,係等於 $x$ 減3的square,呢個又係經常做會錯架啫,咁仲有呢個減號變左加號呢?

S:我諗呢個自己睇錯野啦!

R:睇錯左,點解你咁多睇錯呢?

S:我成日都睇錯野架!

R:抑或係計數的時候唔係好專心,定係諗緊第二D野分心?

S:好似計呢份果陣時呢係recess呢,咁咪就咁寫完佢算囉,可以...

R:諗住休息。

S:又唔計分呢,咪,即係,無乜特別去留意佢囉。

R:唔計分先至反映個真正情況,咁你先至真係做,你係咁樣做出來,即係你最快個反應,你即時果個反應就係做呢樣野出來,即係代表左



呢 ...

S:咁我地就成日計錯數。

R:即係代表左你一唔係好專心的時候呢,你就多呢D咁的錯處,咁所以你可能,第日都要留心下呢計錯數,因為你自己都知道好唔抵。

你個同學頭先又係咁講,你地班成日都計錯數。你好多野識架喎,都係,就係呢D咁樣唔小心,不過我宜家主要就係睇下你地點解計錯數,搵出個原因。即係好多,如果你來講,好明顯就係,你自己就話係一個唔小心啦,但係唔小心我諗都有理由的,就可能會係果陣時諗緊第二D野,或者出面嘈,定點樣,所以你就會寫錯第二D野,可能諗左第二D野,我估係咁,唔知對唔對,你覺得對唔對,係咪呀你覺得?

S:Recess,咁點都會嘈,同埋做呢D卷,你一做完就逐個交呢,唔係做晒先至一齊交呢,咁做完果陣時D人係度嘈呢,咁你咪,係囉,即係照寫囉!

R:主要,如果計分就可能好D咁解。如果計分,係會點好D呀?

S:計分,咁,D人靜D呢又唔會點嘈呢,咁同埋,知道計分咁點都會做好D囉!

R:咁好呀,唔該晒你。

S:唔該。

#### Protocol Analysis of S7

R:好似宜家我地就 paper one 第一題啦。呢度呢,  $\log 5$  加  $\log 2$  變成  $\log 10$  over 2 加  $\log 2$ , 對的, 跟住呢度呢就變成  $\log 1$  over 0.301 乘 0.301, 呢度係點解計到出來呢?

S:因為呢我當時做果陣時呢就見到呢度  $\log 2$  呢, 係等如 0.3010 呀嘛, 咁我咪咁樣答囉。

R:呢個就係  $\log 2$  勒, 咁 10,  $\log 10$  就等如 1 啦喎, 咁  $\log 2$  又變左 0.301, 咁但係宜家個  $\log$  已經用左啦喎!

S:唔需要個  $\log$ .

R:唔需要勒。你宜家睇起上來, 你頭先話比我聽,  $\log 10$  就等如 1 啦, 寫 1 啦, 寫 1 之後仲需要唔需要寫  $\log$ ? 但係你仍然用番  $\log$  係度喎, 點解呢?

S:因為我果陣時唔記得左 D  $\log$  點做呀。

R:唔記得左, 所以就係擺個  $\log$  係呢度勒。

跟住呢, 呢度就係  $\log 64$  變成  $5 \log 2$ , 呢度有問題勒呢度, 呢度點解?

S:因為呢度個  $\log$  乘番佢係, 即係 2 的 5 次方係等如 64 呀嘛。

R:係。呢個呢就係唔識做?

S:係呀, 唔識點做呀!

R:好,跟住就到  $\log 60$  減  $\log 6$ ,呢度呢係  $60$  就變成爲  $3 \cdot 2 \cdot 10$ ,都對播。  
 $\log$  減  $\log$  你就變成爲  $\log$  除  $\log$ ,呢個有無問題呢,抑或係果時你覺得?

S:我做果陣時,我記得係咁做囉。

R:記得  $\log$  減  $\log$  就變成  $\log$  除  $\log$ ,你記得係咁樣,點解你記得呢個方式呢?你實情學左個正確D架,點解記得呢個多過剛才果個?

S:做果陣時係,做果陣時係記得係咁嘛。

R:即係對果D就唔記得左,你淨係記得錯果D啲。

呢個錯架,你知道嘛?

S:我知呢個,計落去果時係記得,因爲你係呢度,即係乘埋係即係  $\log$  乘  $\log$ ,即係  $\log$  乘  $\log$  係等如果舊呀嘛!

R: $\log$  一揸野乘埋就等如  $\log$  相加呀嘛,咁但係如果係減就變左做除。

S:呢個又係一樣啦,係咪, $\log$  減  $\log$  就變成  $\log$  除  $\log$ ,同一樣啲,你覺呢個對的。

R:係呀。

跟住呢度  $\log$  除  $\log$  就變成  $\log$  減  $\log$ ,掉番轉頭,你覺得係對的。

S:因爲做果陣時覺得,如果呢度,係減變左除啦,咁肯定你除,掉番轉變番做減。

R:如果係咁,問你勒,有時呢個  $\log$  減呢個  $\log$ ,唔係,呢個減呢個,變唔變成呢個除呢個呢?

S:唔變成。

R:都係有分別的。

好啦,跟住呢度呢, $\log 100$  除  $10$  你就變成  $2 \text{ over } 1$ ,呢個點解呀?

S:因爲,拿,係咁,即係當佢呢個  $\log$  係兩個都有份,咁  $\log 100$  係  $2$ ,  $\log 10$  係  $1$  囉咪。

R:即係變左  $\log 100 \text{ over } \log 10$ ,呢個第九題。

跟住呢就係  $\log 3 \text{ over } 5$  等如呢個就對啦, $2, \log 6$  減  $\log 10$ ,係試下解釋呢行比我,呢行解呀?呢個第十二題,就  $\log 6 \text{ over } \log 12$  加  $\log 2$ ,就變成爲呢,呢個子就係  $\log 6$  減  $\log 10$ ,分母就  $\log 12$  加  $\log 2$ 。

S:咁呢度係  $\log$ ,咁  $\log 6 \text{ over } 10$  咁即係可以變左做減呀,咁跟住呢度,咁減又可以變除囉。

R:但係呢個變左除的時候,呢個需唔需要變呢?

S:呢個,要呀。

R:不過你果陣時就無做。

S:即係果陣時無諗得咁詳細。

R:即係都係果個...



跟住第十四題,  $\log 10$  乘  $10 \text{ over } 10$  加  $10$ , 就變成  $\log 10$  加  $\log 10$ , 跟住下面  $\log 10$  加  $\log 10$ , 將個  $\log$  分散佢呀, 係嘛? 但係你睇下有D奇怪,  $\log 10$  乘  $10$  就變成  $\log 10$  加  $\log 10$ , 咁  $\log 10$  加  $10$ , 你又變成  $\log 10$  加  $\log 10$ .

S: 呢個乘黎架!

R: 哦, 即係乘就變做加, 加又變左乘。

S: 係, 係。

R: 咁到呢度呢?

S: 呢度又好似呢度咁囉, 因為除呀, 咁可以  $\log 7$  減  $\log 9$  加跟住  $\log 3$  呀, 減番, 因為呢度係除, 咁減番  $\log 28$  囉, 咁跟住呢個減呀嘛, 咁即係變左係下底。

R: 哦, 減就變左除。

S: 係呀。

R: 減又變左除, 好得意呀呢個, 即係你覺得減就可以變做除的, 加就變左乘, 掉番轉又係咁, 係咪所有情況都係咁呢?

S: 唔係。

R: 幾時可以變做減?

S: 要睇番條題目, 即係條題目係加變左減, 咁咪計到囉, 唔駛用呢個囉。即係諗過係度, 變左係計得到的, 咁就用果個囉! 如果係計唔到呢, 咪諗過第二個囉又。

R: 即係計得就用佢, 計唔到就, 即係複雜D, 就唔計落去勒, 咁解。

好, 跟住就 paper two 第一題, 拿, 呢個呢, 呢度來到呢度呢, 即係  $\log$  開方  $x$  over  $x$ .

S: 呢個  $x$  可以寫做二分一架嘛, 如果將個  $\log$  擺, 即係可以將二分一擺係前面, 變左二分一個  $\log x$ , 咁呢個係除變左可以減  $\log x$ .

R: 呢個錯架下, 等陣話番比你聽。呢度呢第三題。

S: 第三題因為你個 square 寫左係度啦, 可以擺係前面啦, 咁擺左係前面, 咁減兩個可以變左除, 咁兩個咪可約得囉大家。

R: 我想問下頭先個問題, 即係例如話, 如果變左做複雜D, 就唔想做落去, 即係, 宜家減變左除, 拿, 咁唔係好做架啦, 你就唔做落去勒。

S: 係呀。

R: 跟住到呢度勒, 係 section B 第一題,  $\log x$  加  $6$  加  $1$  等如  $0$ , 變成  $\log 6x$  加  $1$  等於  $0$ , 呢個  $\log 6x$  點來?

S: 因為係呢度  $\log$  呢, 佢加咁即係乘囉, 跟住大家, 佢抽個  $\log$  出來, 即係大家抽係  $\log$ , 咁兩個乘埋佢咪變左  $\log 6x$  囉。

R: 跟住呢度呢,  $\log 6x$  等於  $\log$  十分一, 呢度無問題。跟住就到第四,  $\log x$  square 等如  $\log 10$ , 呢個點解呢?



S:因爲呢度又係括弧加,咁即係當乘。

R:乘入去,即係  $x^2$  乘 1,即係話宜家變成  $\log x^2$  乘 1 等於 1,就係因爲要快番 D,如果唔係就做唔到。

S:如果唔係要諗好耐囉。

R:你,當你咁做的時候,你覺得對唔對先?

S:我覺得對架!

R:覺得對,見到簡單就對了。

跟住呢就到第五題,  $\frac{1}{3} \log x + 1$  就變成  $\frac{1}{3} \log x - \frac{1}{2}$ 。

S:正確呀。

R:跟住就  $\frac{1}{3} \log x - \frac{1}{2} \log x$  就變成  $\frac{1}{3} \log x$  over  $\frac{1}{2} \log x$ , 又係因爲咁樣?

S:都係減。

R:減就變做除,原因就是係可以入左去。

但係呢個係 solve equation 喎,你留意下你係咪 solve equation;  $x$  係等於乜野,你無留意搵出  $x$ ,你約左去。

S:因爲你就咁寫,無寫係搵  $x$  囉,咁就咁諗住約簡佢。

R:呢度又係咁樣,呢度又係無左  $x$  等於咩野。跟住呢度就係第六題,  $\log x^2 - 3 \log x^2 - 9 + 1 = 0$ ,你就變左係,左面無左野,等於右面,呢度是兩個分數,點解呢?

S:咁呢度又係減,  $\log x^2 - 3$  即係又係  $\log x^2$  over 3,跟住呢度又減 又 over 番呢度,咁跟住加 1。

R:咁加 1 比埋左手面,唔比右手面喎,加 1 需唔需要變?

S:我做果時...因爲我呢度係唔夠時間做,呢度係比較後 D 先做,唔係好夠時間。

R:宜家識唔識做呢?

S:宜家識。

R:宜家識咁唔駛理佢啦。咁好啦,唔該晒你先。

=====  
Note: R=Researcher; S= Student

## Appendix S

### *Installation and Source Code*

The software Electronic Homework and the Source Codes can be found in the set of disks included. The first set contains the installation disks of Electronic Homework while the second set contains the source codes, which are stored in two separate files according to the purposes of the codes. The first file, called the “Logex.txt”, stores the Prolog codes that are used in the Expert, Student and Tutoring Modules. On the other hand, the other file, called the “VBInter.txt”, contains the codes that are responsible for the interface between human user and the computer, that is, this file contains the codes used in the Communication Module. The following shows how Electronic Homework can be installed into a computer system and how the source code can be inspected.

#### **How to install Electronic Homework:**

Configuration requirement: A 80486 or higher PC compatible computer with a hard drive and a floppy drive is usually required, although a fast machine would be an advantage. Chinese Windows version 3.1 is necessary to use Electronic Homework. You should also have a formatted disk at drive A to store the data.

Installation procedure:

Boot the computer and log in Chinese Windows. Place Disk 1 of the setup disks into drive A.

From the Program Manager, choose “File” and then “Run”. From there, type in “a:setup.exe”.

Follow the instructions shown on the screen.

#### **How to Read the Source Code**

Two files, called the “Logex.txt” and the “VBInter.txt”, in Disk 3 are used to store the Prolog codes and Visual Basic codes. The Prolog codes are used in the Expert, Student and Tutoring Modules and are representing the knowledge of the system. On the other hand, the Visual Codes are responsible for interfacing human user and the computer. Both are in the form of text files and can be read by using any text editors or word processors.





CUHK Libraries



003511141